



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Bachelorarbeit

Vergleich von Ansätzen zur numerischen Lösung von
MPCCs am Beispiel eines Mautproblems

Vorgelegt an der Professur für Numerische Mathematik (Partielle
Differentialgleichungen) von Prof. Dr. Herzog

Name: Eric Legler
Matrikelnummer: 277868
Betreuer: Prof. Dr. Roland Herzog, Dr. Gerd Wachsmuth
Erstkorrektor: Prof. Dr. Roland Herzog
Zweitkorrektor: Dr. Gerd Wachsmuth

Abstract

In der folgenden Arbeit werden spezielle Mautoptimierungsprobleme betrachtet. Diese stellen Bilevelprobleme dar, welche als Problem der unteren Stufe ein Mehrgüterflussproblem besitzen. Im Problem der oberen Stufe sollen die verbleibenden Restkapazitäten auf dem zugrundeliegenden Netzwerk an eine vorgegebene Verteilung dieser angenähert werden, indem auf den Kanten des Netzwerkes Mautgebühren erhoben werden. Die betrachteten Optimierungsprobleme werden in MPCCs ungeschrieben und in der von Hatz et al. in [9] vorgeschlagenen Weise modifiziert. Dabei fällt auf, dass die Hauptabsicht dieser Modifikation, welche die Gültigkeit der MPCC-LICQ im Problem der oberen Stufe ist, nicht erfüllt wird. Für den numerischen Vergleich werden die von Hoheisel, Kanzow und Schwartz in [23] vorgestellten Regularisierungsverfahren für MPCCs in MATLAB implementiert. Die erhaltenen Resultate zeigen, dass das Regularisierungsverfahren von Scholtes, bei der Berechnung der hier genutzten selbst erstellten Beispielprobleme, die besten Resultate liefert, unabhängig davon ob die Idee aus [9] verwendet wird oder nicht. Generell wurden bessere Ergebnisse erzielt, wenn die Modifikation von Hatz nicht verwendet wurde.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
Tabellenverzeichnis	7
Übersicht der verwendeten Algorithmen	8
1. Einleitung	9
2. Modellierung des Mautproblems	10
2.1. Modellierung des Problems der unteren Stufe	10
2.2. Modellierung des Problems der oberen Stufe	13
2.2.1. Formulierung ohne Lift	15
2.2.2. Formulierung mit Lift	17
2.3. Beispiele	19
3. Untersuchung auf Constraint Qualifications	22
3.1. Definitionen und Vorbetrachtungen	22
3.2. Formulierung ohne Lift	25
3.3. Formulierung mit Lift	30
4. Regularisierungsverfahren	36
5. Beispielprobleme	39
6. Numerische Betrachtungen	40
6.1. Grundlegende verwendete Algorithmen	40
6.1.1. Algorithmen für die Regularisierungen	40
6.1.2. Umsetzung des Lifts	40
6.2. Ergebnisse der vorgestellten Beispiele	42
6.3. Resultate der in Kapitel 5 generierten Beispiele	43
7. Diskussion der Ergebnisse	46
8. Conclusion	50
Literatur	51
A. Herleitung der Lösungen von Beispiel 2.2	54

B. Algorithmus zur Erstellung der Beispielprobleme	56
C. Algorithmen zur Lösung regularisierter Probleme	59
D. Ergebnisse der vorgestellten Beispiele	61
E. Ergebnisse bei modifizierten Beispiele	63

Abbildungsverzeichnis

1.	Beispielgraph 2.1	19
2.	Beispielgraph 2.2	21
3.	Vergleich der berechneten Zielfunktionswerte	44
4.	Vergleich der Verletzung der Nebenbedingungen	44
5.	Vergleich der benötigten Zeit	45
6.	Vergleich der Verletzung der Nebenbedingungen	47
E.1.	Vergleich der berechneten Zielfunktionswerte	63
E.2.	Vergleich der Verletzung der Nebenbedingungen	65
E.3.	Vergleich der benötigten Zeit	65

Tabellenverzeichnis

1.	Beispieldaten 2.1	20
2.	Beispieldaten 2.2	21
3.	Beispielprobleme	39
A.1.	Mögliche Pfade	54
A.2.	Optimaler Fluss	54
C.1.	Übersicht der verwendeten Parameter	59
D.1.	Ergebnisse von Beispiel 2.1	61
D.2.	Ergebnisse von Beispiel 2.2	62

Übersicht der verwendeten Algorithmen

1. Algorithmus zum Lösen der gelifteten Probleme	41
B.1. Erstellung der Beispielprobleme	56
C.1. Allgemeiner Algorithmus zum Lösen der reg. Probleme	59
C.2. Verbesserter Algorithmus zum Lösen der reg. Probleme	60

1. Einleitung

In den letzten Jahren spielten Überlegungen zur Gestaltung eines deutschen Mautsystems wiederholt eine Rolle. So wurde im Juli 2011 eine Maut für nahezu alle Kraftfahrzeuge über 12 Tonnen eingeführt, die bei der Benutzung von Autobahnen und größeren Bundesstraßen entrichtet werden muss [3, §1 Abs.1,2]. Eine Ausweitung dieser Maut auf alle Bundesstraßen und darüber hinaus auch die Bemaßung von Fahrten mit einem PKW wurde im Koalitionsvertrag [4, S. 29] der Regierung der 18. Legislaturperiode festgelegt. Das für LKW existierende Mautsystem sieht vor, dass für jeden genutzten Straßenkilometer ein festgesetzter Geldbetrag gezahlt werden muss. In [18] wurde eine Überlegung vorgestellt, nach der die Höhe der Maut von der Auslastung der Straße abhängig gemacht wird und damit der Verkehrsfluss auf einem Straßennetzwerk gesteuert werden kann. Grundlage dafür ist, dass sich die einzelnen Verkehrsteilnehmer in dem Sinne optimal verhalten, dass sie jeweils den Weg wählen, der für sie am kostengünstigsten ist. Der Betreiber eines solchen Mautsystems kann also durch das Anheben der Maut auf einem stark frequentierten Streckenabschnitt erreichen, dass die Verkehrsteilnehmer einen anderen, kostengünstigeren Weg wählen, falls ein solcher existiert. In diesem Fall sinkt folglich das Verkehrsaufkommen auf dem Abschnitt mit der gestiegenen Maut. Mathematisch gesehen handelt es sich bei einem solchen Problem um ein Bilevelproblem. Das Problem der unteren Stufe stellt dabei das optimale Verhalten der einzelnen Verkehrsteilnehmer dar, während das Problem der oberen Stufe die Optimierung des Verkehrsflusses durch den Betreiber des Mautsystems repräsentiert. Formuliert man das Problem der unteren Stufe um, erhält man ein mathematical problem with complementary constraints (MPCC), also ein Optimierungsproblem mit Komplementaritätsbedingungen. Da diese Probleme numerisch schwierig zu behandeln sind, wurden in [23] verschiedene, in letzter Zeit entstandene Regularisierungsverfahren vorgestellt und diese sowohl vom Standpunkt der theoretischen Eigenschaften als auch der numerischen Ergebnisse verglichen. Gegenstand dieser Regularisierungsverfahren ist das Abschwächen der Komplementaritätsbedingungen. Durch stufenweises Verstärken dieser neuen Bedingungen und somit dem Lösen einer Vielzahl von Optimierungsaufgaben soll letztendlich die Lösung des ursprünglichen Problems numerisch leichter möglich sein. Ein anderer Ansatz wird in [9] verfolgt. Hier werden durch ein Modifizieren des Problems bessere theoretische Eigenschaften erreicht. Somit soll das Problem besser lösbar sein, ohne das Auftreten von Komplementaritätsbedingungen zu umgehen. In der vorliegenden Arbeit soll untersucht werden, welcher der beiden Ansätze bei der Anwendung auf Mautprobleme bessere Resultate liefert und ob eine Kombination beider Konzepte die Ergebnisse weiter verbessern kann.

2. Modellierung des Mautproblems

Wie in der Einleitung erwähnt, handelt es sich bei dem vorliegenden Problem um ein sogenanntes Bilevelproblem, also ein mehrstufiges Optimierungsproblem. Für die Modellierung wird nun zunächst das Problem der unteren Stufe betrachtet. Anschließend wird das Problem der oberen Stufe ausgearbeitet. Der wichtigste Aspekt dabei ist das Formulieren von Nebenbedingungen, die zur Optimalität des Problems der unteren Stufe äquivalent sind.

2.1. Modellierung des Problems der unteren Stufe

Das Problem der unteren Stufe stellt sich als ein Mehrgüterflussproblem dar, welches beispielsweise in [16, Kapitel 17] thematisiert wird. Diese Optimierungsprobleme sind dadurch gekennzeichnet, dass bestimmte Mengen mehrerer Güter gleichzeitig auf einem gemeinsamen Netzwerk transportiert werden. Das zugrunde liegende Netzwerk soll an dieser Stelle Straßen zwischen verschiedenen Orten repräsentieren, während die Flüsse der einzelnen Güter Wege darstellen, welche jeweils durch eine bestimmte Anzahl an Fahrzeugen befahren werden. Die Fahrzeuge, welche jeweils am selben Ort starten und das selbe Ziel haben, werden dabei zu Gruppen zusammengefasst und im Weiteren durch ein Gut repräsentiert. Damit ergibt sich, dass die Menge eines Gutes gleich der Anzahl der Fahrzeuge der zugehörigen Gruppe ist. Für die Modellierung soll im Folgenden die Annahme gelten, nach der sich die Gruppen der Fahrzeugnutzer in dem Sinn rational verhalten, dass die Gesamtkosten der Fahrer für die Nutzung der Straßen minimiert werden. Diese Annahme wurde auch in der Modellierung von Dempe in [18] so getroffen.

Der Hauptaspekt bei der weiteren Formulierung dieser Probleme als Optimierungsaufgabe ist die Herleitung der Nebenbedingungen, die sich als Folge des Transportes auf dem Straßennetzwerk ergeben. Dafür werden an dieser Stelle drei Definitionen eingeführt, die sich jeweils in [16, Kapitel 2] beziehungsweise [13, Abschnitt 9.1] wiederfinden.

Definition 2.1. Ein gerichteter Graph $D = (N, A)$ besteht aus einer Knotenmenge N und einer Menge A von Bögen, welche aus geordneten Paaren verschiedener Knoten besteht. Weiterhin heißt Graph $D' = (N', A')$ Teilgraph von D , falls $N' \subset N$ und $A' \subset A$ gilt.

Die folgenden beiden Definitionen werden ebenfalls im weiteren Verlauf benötigt und sollen deshalb schon an dieser Stelle formuliert werden.

Definition 2.2. Ein gerichteter Kantenzug ist ein Teilgraph von D , der aus einer Folge von Knoten und Bögen $i_1 - a_1 - i_2 - a_2 - \dots - i_{r-1} - a_{r-1} - i_r$ besteht, wobei $1 \leq k \leq r-1$ und $a_k = (i_k, i_{k+1}) \in A$ oder $a_k = (i_{k+1}, i_k) \in A$ gilt. Tritt in einem gerichteten

Kantenzug keine Wiederholung von Knoten auf, so spricht man von einem gerichteten Pfad.

Definition 2.3. Ein Graph heißt zusammenhängend, wenn jedes Paar (i, j) seiner Knoten zusammenhängend ist, also mindestens ein Pfad von i nach j existiert. An dieser Stelle genügt ein ungerichteter Pfad, der wie ein gerichteter Pfad definiert ist, gleichzeitig jedoch die Richtung der Bögen nicht beachtet wird.

Im Weiteren soll nun $n = |N|$ und $m = |A|$ gelten. Eine Ordnung auf den Mengen N und A wird ebenfalls vorausgesetzt. Mit K soll die Menge der Fahrzeuggruppen bezeichnet werden und darüber hinaus wird $k = |K|$ gesetzt.

Auch wenn die folgende Modellierung mittels gerichteter Graphen stattfindet, wird weiterhin vereinfachend nur noch von Graphen gesprochen. Dabei sind aber dennoch stets gerichtete Graphen gemeint. Dies gilt auch für die gerichteten Pfade.

Analog zu den Bezeichnungen beispielsweise in [12] werden im Folgenden die Knoten, die Ausgangspunkt für den Transport eines Gutes sind, als *Quellen* und die Knoten, welche die Zielorte darstellen, als *Senken* bezeichnet. Aufgrund der Tatsache, dass jedes Gut eine Gruppe von Fahrzeugen repräsentiert, die jeweils den selben Startknoten und gleichzeitig den selben Zielknoten haben, hat jedes Gut genau eine Quelle und genau eine Senke.

Nun wird dargestellt, welche Nebenbedingungen sich aus den Eigenschaften des Straßennetzes sowie den Transporten auf diesen ergeben.

- Knotenbilanzen

Als Knotenbilanz bezeichnet man die Differenz zwischen der Summe der im Knoten eingehenden Ströme und der Summe der vom Knoten ausgehenden Ströme. Als Nebenbedingung für das Optimierungsproblem müssen nun für jede Gruppe $l \in K$ die Knotenbilanzen wie folgt gesetzt werden:

- gleich der Gruppengröße mit negativen Vorzeichen in der Quelle,
- gleich der Gruppengröße mit positivem Vorzeichen in der Senke
- und für alle weiteren Knoten gleich 0.

- Einhaltung der Kapazitätsobergrenzen

Für alle Kanten $(i, j) \in A$ muss die Summe der über diese Kante stattfindenden Ströme aller Waren kleiner als die Kapazität dieser Kante sein.

- Nichtnegativität der transportierten Menge

Für alle Waren $l \in K$ muss der Strom auf allen Kanten $(i, j) \in A$ nichtnegativ sein.

Diese Bedingungen werden nun in Gleichungen und Ungleichungen gefasst. Dazu wird zunächst b_j^l für alle Güter $l \in K$ und alle Knoten $j \in N$ definiert:

$$b_j^l := \begin{cases} -v_{qs}^l, & \text{für } j = q \\ v_{qs}^l, & \text{für } j = s \\ 0, & \text{sonst.} \end{cases}$$

Hierbei wird für jeden Transport l mit v_{qs}^l die Menge der zu transportierenden Waren vom der Quelle q zur Senke s bezeichnet.

Im Weiteren soll mit $x_{i,j}^l$ die Zahl bezeichnet werden, die angibt, welche Menge von Gut l auf der Kante (i,j) fließt. Diese Größe wird auch als Strom des Gutes l auf der Kante (i,j) bezeichnet. Analog zu [18, Formel (7.3)] ergeben sich damit für den Fluss auf dem Graphen D folgende Nebenbedingungen:

$$\begin{aligned} \sum_{(i,j) \in E} x_{i,j}^l - \sum_{(j,i) \in E} x_{j,i}^l &= b_j^l && \text{für alle } j \in N \text{ und } l \in K \\ \sum_{l=1}^k x_{i,j}^l &\leq u_{i,j} && \text{für alle } (i,j) \in A \\ x_{i,j}^l &\geq 0 && \text{für alle } (i,j) \in A \text{ und } l \in K \end{aligned} \quad (2.1)$$

Die Darstellung dieser Nebenbedingungen soll nun vereinfacht werden. Dazu wird mit der folgenden Definition aus [12, Abschnitt 2.3, S.29] der Begriff der Inzidenzmatrix eingeführt.

Definition 2.4. Als Inzidenzmatrix eines gerichteten Graphen $D = (N, A)$ wird die Matrix $\mathcal{N} = (a_{r,(i,j)})_{r \in N, (i,j) \in A}$ mit

$$a_{r,(i,j)} = \begin{cases} -1, & \text{für } r = i \\ 1, & \text{für } r = j \\ 0, & \text{sonst} \end{cases}$$

bezeichnet.

Somit gilt insbesondere $\mathcal{N} \in \mathbb{R}^{n \times m}$. Wie später in Lemma 3.1 gezeigt wird, ist es an dieser Stelle notwendig eine beliebige Zeile aus der Inzidenzmatrix \mathcal{N} herauszustreichen, damit sich die Eigenschaften des Problems verbessern. Sei nun q die Nummer einer beliebigen Zeile aus \mathcal{N} oder dazu äquivalent $q \in N$. Die Äquivalenz folgt dabei leicht aus der Überlegung, dass die Zeilen von \mathcal{N} die Knoten N repräsentieren. Im Weiteren sei nun $\hat{\mathcal{N}} \in \mathbb{R}^{(n-1) \times m}$ die Matrix, die man erhält, wenn man aus \mathcal{N} die Zeile q weglässt.

Weiterhin gelten folgende Bezeichnungen:

$$\begin{aligned} \bar{\mathcal{N}} &= \begin{pmatrix} \hat{\mathcal{N}} & & 0 \\ & \ddots & \\ 0 & & \hat{\mathcal{N}} \end{pmatrix} \in \mathbb{R}^{k(n-1) \times km}, & J &= (I \ \cdots \ I) \in \mathbb{R}^{m \times km}, \\ x^l &= (x_{i,j}^l)_{(i,j) \in A} \quad l \in K, & b^l &= (b_j^l)_{j \in N \setminus \{q\}} \quad l \in K, \\ x_u &= (x^1 \ \cdots \ x^k)^\top \in \mathbb{R}^{km}, & b &= (b^1 \ \cdots \ b^k)^\top \in \mathbb{R}^{k(n-1)}, \\ u &= (u_{i,j})_{(i,j) \in A}. \end{aligned} \quad (2.2)$$

Setzt man diese Bezeichnungen in (2.1) ein, werden die Nebenbedingungen wie gewünscht vereinfacht und man erhält eine Problemformulierung, die Formel (17.1) aus [16] ähnlich ist.

Nun soll die Zielfunktion des Problems der unteren Stufe betrachtet werden, also die zu optimierende Größe gewählt werden. Wie schon erwähnt, soll analog zu [18] ein gesamtkostenminimaler Fluss betrachtet werden. Es wird also angenommen, dass sich jede Gruppe von Fahrzeugen so verhält, dass die Summe der Kosten für alle Fahrer minimal wird.

Die Kosten pro Kante setzen sich wie in [18] jeweils aus den mautunabhängigen Kosten $c \in \mathbb{R}^m$ und den durch die Maut verursachten Kosten $c_m \in \mathbb{R}^m$ zusammen. Da die Maut im Problem der oberen Stufe optimiert werden soll, wird c_m an dieser Stelle mit in die Modellierung des Problems der unteren Stufe einbezogen, allerdings zunächst als konstant angesehen. In dieser Modellierung sollen c und c_m für alle Transporte identisch sein. Dies wird umgesetzt, indem beide Vektoren von links mit J^\top multipliziert werden. Dadurch erhält man die Zielfunktion des Problems der unteren Stufe.

Insgesamt lautet das Problem der unteren Stufe:

$$\begin{aligned}
 & \underset{x_u \in \mathbb{R}^{km}}{\text{minimiere}} && (c + c_m)^\top J x_u \\
 & \text{unter} && \bar{N} x_u = b \\
 & && J x_u \leq u \\
 & && x_u \geq 0
 \end{aligned} \tag{2.3}$$

2.2. Modellierung des Problems der oberen Stufe

Nun soll das Problem der oberen Stufe modelliert werden, wobei auch hier die Modellierung in Anlehnung an [18] erfolgte. In den Nebenbedingungen dieses Problems kommen zu der Optimalität des Problems der unteren Stufe zusätzlich auch Bedingungen an c_m hinzu. Während in [18] auf die möglichen Nebenbedingungen an c_m nicht detailliert eingegangen wurde, sollen diese in diesem Modell durch eine untere Schranke l_{c_m} und eine obere Schranke u_{c_m} dargestellt werden. Damit auch Mautvektoren berücksichtigt werden können, bei denen es Komponenten gibt, die keine obere oder untere Schranke besitzen, werden für die Komponenten von l_{c_m} und u_{c_m} auch die Werte $+\infty$ beziehungsweise $-\infty$ zugelassen.

Wie schon in der Einleitung erwähnt, soll mit der Maut der Verkehrsfluss gesteuert werden. In [18] wurde dies dadurch umgesetzt, dass der Betreiber des Mautnetzwerkes eine spezielle Verteilung der Restkapazitäten x_z auf dem Netzwerk vorgibt und dann die euklidische Norm der Differenz aus den tatsächlichen Restkapazitäten und x_z gebildet

wird. Insgesamt erhält man damit als Problem der oberen Stufe:

$$\begin{aligned}
 & \underset{\substack{c_m \in \mathbb{R}^m \\ x_u \in \mathbb{R}^{km}}}{\text{minimiere}} && \|u - Jx_u - x_z\| \\
 & \text{unter} && c_m \geq l_{c_m} \\
 & && c_m \leq u_{c_m} \\
 & \underset{x_u \in \mathbb{R}^{km}}{\text{minimiere}} && (c + c_m)^\top Jx_u \\
 & \text{unter} && \bar{N}x_u = b \\
 & && Jx_u \leq u \\
 & && x_u \geq 0.
 \end{aligned} \tag{2.4}$$

Zu beachten ist bei dieser Modellierung jedoch, dass diese eine weitere Annahme an das Verhalten der Nutzer des Straßennetzwerkes voraussetzt. Liegt der Fall vor, dass es mehrere mögliche Flüsse mit identischen Gesamtkosten gibt, so wird in dieser Modellierung stets der Fluss gewählt, der die Zielfunktion des Problems der oberen Stufe minimiert. Diese Problematik wurde auch in [18] thematisiert.

Wie schon zu Beginn dieses Kapitels angedeutet, soll nun das Problem der unteren Stufe ersetzt werden. Dazu soll nun zunächst analog zu [2, Formel (4.32)] das allgemeine Optimierungsproblem

$$\begin{aligned}
 & \underset{x \in \mathbb{R}^p}{\text{minimiere}} && f(x) \\
 & \text{unter} && Bx \leq b \\
 & && Cx = c
 \end{aligned} \tag{2.5}$$

betrachtet werden, wobei b und c Vektoren bzw. B und C Matrizen von der jeweils passenden Dimension sowie f eine Funktion $f : \mathbb{R}^p \rightarrow \mathbb{R}$ sind. Es gelten die nun folgenden Definitionen.

Definition 2.5. (vgl. [27, S. 274])

Sei $f : \mathbb{R}^p \rightarrow \mathbb{R}$ eine Funktion. f heißt konvex, falls für alle $x, y \in \mathbb{R}^p$ und alle $\lambda \in [0, 1]$ gilt:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Im folgenden Satz werden die Karush-Kuhn-Tucker(KKT) Bedingungen eingeführt, die später zum Umschreiben des Problems der unteren Stufe verwendet werden sollen.

Satz 2.1. (vgl. [2, Theorem 4.11])

a) Sei x^* ein lokales Minimum von (2.5), dann gibt es Vektoren λ^* und μ^* von passender Größe, sodass die folgenden KKT-Bedingungen erfüllt sind:

$$\nabla f(x^*) + B^\top \mu^* + C^\top \lambda^* = 0 \quad (2.6a)$$

$$Cx = c \quad (2.6b)$$

$$0 \leq \mu \perp b - Bx \geq 0, \quad (2.6c)$$

wobei für zwei Vektoren $a, b \in \mathbb{R}^q$ die Bedingung $a \perp b$ bedeutet, dass $a_i b_i = 0$ für alle $i = 1, \dots, q$.

b) Ist $f(x)$ konvex und existieren Vektoren x^+, λ^+ und μ^+ , die die KKT-Bedingungen (2.6) erfüllen, dann ist x^+ ein globales Minimum von (2.5).

Mit diesen Resultaten soll das nun folgende Korollar bewiesen werden.

Korollar 2.2. *Ist die Funktion $f(x)$ aus (2.5) linear, so ist x^* genau dann eine globale Lösung von (2.5), wenn es Vektoren λ^* und μ^* der jeweils passenden Größe gibt, sodass x^*, λ^* und μ^* die KKT-Bedingungen (2.6) erfüllen.*

Beweis. „ \Rightarrow “ Sei x^* eine globale Lösung von (2.5) und somit auch eine lokale Lösung. Die Anwendung von Punkt a) aus Satz 2.1 liefert, dass es Vektoren λ^* und μ^* gibt mit denen (2.6) erfüllt ist.

„ \Leftarrow “ Seien nun x^*, λ^* und μ^* Vektoren, sodass (2.6) erfüllt ist. Da $f(x)$ linear ist, gibt es einen Vektor c , sodass $f(x) = c^\top x$. Somit gilt insbesondere $f(x) = c^\top c = \lambda c^\top x + (1 - \lambda)c^\top x = \lambda f(x) + (1 - \lambda)f(x)$ für alle $\lambda \in [0, 1]$. Folglich ist $f(x)$ konvex und die Anwendung von Teil b) aus Satz 2.1 liefert, dass x^* ein globales Minimum von (2.5) ist. □

Anhand dieses Korollars wird auch deutlich, warum die Umformulierung der dargestellten Bilevelprobleme zu MPCCs führt, da durch die Nebenbedingung (2.6c) eine solche Komplementaritätsbedingung in den Nebenbedingungen des Problems der oberen Stufe auftreten wird.

In dieser Arbeit soll ein numerischer Vergleich zwischen der in [9] vorgeschlagenen Modifikation und der Formulierung ohne eine derartige Veränderung erfolgen. Zunächst wird die Formulierung ohne Lift betrachtet.

2.2.1. Formulierung ohne Lift

Nun soll das unmodifizierte Problem der unteren Stufe durch seine KKT-Bedingungen ersetzt werden. Dazu wird das Problem (2.3) zunächst in eine Form analog zu (2.5) gebracht.

$$\begin{aligned} & \underset{x_u \in \mathbb{R}^{km}}{\text{minimiere}} && (J^\top(c + c_m))^\top x_u \\ & \text{unter} && \begin{pmatrix} J \\ -I \end{pmatrix} x_u \leq \begin{pmatrix} u \\ 0 \end{pmatrix} \\ & && \bar{N}x_u = b \end{aligned}$$

Hierbei ist der Vektor 0 von der passenden Dimension. An dieser Stelle wird deutlich, dass das Korollar 2.2 auf dieses Problem angewendet werden kann und somit das Problem der unteren Stufe äquivalent durch seine KKT-Bedingungen ersetzt werden kann. Die KKT-Bedingungen (2.6) für das Problem der unteren Stufe lauten:

$$\begin{aligned} J^\top(c + c_m) + \bar{N}^\top \lambda + J^\top \mu_1 - \mu_2 &= 0 \\ Ax_u - b &= 0 \\ 0 \leq \mu_1 \perp u - Jx_u &\geq 0 \\ 0 \leq \mu_2 \perp x_u &\geq 0. \end{aligned}$$

Setzt man dies in (2.4) ein, erhält man:

$$\begin{aligned} &\text{minimiere} && \|u - Jx_u - x_z\|^2 \\ &\begin{array}{l} c_m \in \mathbb{R}^m \\ x_u \in \mathbb{R}^{km} \\ \lambda \in \mathbb{R}^{k(n-1)} \\ (\mu_1, \mu_2)^\top \in \mathbb{R}^{m+km} \end{array} && \\ \text{unter} &&& \begin{aligned} J^\top(c + c_m) + \bar{N}^\top \lambda + J^\top \mu_1 - \mu_2 &= 0 \\ \bar{N}x_u - b &= 0 \\ c_m &\geq l_{c_m} \\ c_m &\leq u_{c_m} \\ 0 \leq \mu_1 \perp u - Jx_u &\geq 0 \\ 0 \leq \mu_2 \perp x_u &\geq 0 \end{aligned} \end{aligned} \quad (2.7)$$

Wie in [20] empfohlen, wird nun zur Verbesserung der numerischen Resultate in der ersten Komplementaritätsnebenbedingung die Slackvariable s_1 eingeführt. Darüber hinaus alle weiteren Nebenbedingungen, außer dem Komplementaritätssystem und den Schranken von c_m , in Matrixschreibweise formuliert. Zur Vereinfachung der Notation wird weiterhin die Schreibweise

$$\begin{aligned} x_o &= (c_m, x_u, \lambda, \mu_1, \mu_2, s_1)^\top, \text{ wobei} \\ x_o &\in \mathbb{R}^{(2k+3)m+k(n-1)} =: \mathbb{R}^{p_1} \end{aligned}$$

verwendet. Setzt man diese beschriebenen Veränderung im Problem 2.7 um, ergibt sich folgendes Problem:

$$\begin{aligned} &\text{minimiere} && \|u - Jx_u - x_z\|^2 \\ &\begin{array}{l} x_o \in \mathbb{R}^{p_1} \end{array} && \\ \text{unter} &&& \begin{aligned} \begin{pmatrix} J^\top & \bar{N}^\top J^\top & -I \\ \bar{N} & & \\ J & & I \end{pmatrix} x_o &= \begin{pmatrix} -J^\top c \\ b \\ u \end{pmatrix} \\ c_m &\geq l_{c_m} \\ c_m &\leq u_{c_m} \\ 0 \leq \mu_1 \perp s_1 &\geq 0 \\ 0 \leq \mu_2 \perp x_u &\geq 0. \end{aligned} \end{aligned} \quad (2.8)$$

2.2.2. Formulierung mit Lift

Nun soll das Problem nach der Vorschrift, die von Hatz et al. in [9] dargestellt wurde, modifiziert werden. Zunächst folgt eine Betrachtung der Nebenbedingungen des neuen Optimierungsproblems.

In [9, Abschnitt 1] wurde darauf hingewiesen, dass die Nebenbedingungen des Problems der oberen Stufe, die das Problem der unteren Stufe nicht beeinflussen, bei der Modifikation nicht verändert werden müssen. Im vorliegenden Modell sind das die Bedingungen an den Mautvektor c_m .

Somit wird nun zunächst das Problem der unteren Stufe und seine Veränderungen durch den Lift betrachtet. Dieses Problem soll nun in der Form

$$\begin{aligned} & \underset{x_u \in \mathbb{R}^{km}}{\text{minimiere}} && \left(J^\top(c + c_m) \right)^\top x_u \\ & \text{unter} && \begin{aligned} \bar{N}x_u - b &= 0 \\ -Jx_u + u &\geq 0 \\ x_u &\geq 0 \end{aligned} \end{aligned}$$

betrachtet werden. Wie ebenfalls im ersten Abschnitt von [9] erwähnt, werden die Gleichungsnebenbedingungen des Problems der unteren Stufe nicht verändert. Die Ungleichungsnebenbedingungen werden hingegen wie in [9, Formel (2.11)] mit neuen Lifting-Variablen w_i modifiziert, welche im Folgenden stets von der jeweils passenden Dimension sein sollen. Insgesamt erhält man nun

$$\begin{aligned} & \underset{x_u \in \mathbb{R}^{km}}{\text{minimiere}} && \left(J^\top(c + c_m) \right)^\top x_u \\ & \text{unter} && \begin{aligned} \bar{N}x_u - b &= 0 \\ -Jx_u + u &\geq w_1 \\ x_u &\geq w_2. \end{aligned} \end{aligned} \tag{2.9}$$

Dieses Problem, das als Nebenbedingung im Problem der oberen Stufe auftritt, soll nun analog zu den Betrachtungen in [9] durch seine KKT-Bedingungen ersetzt werden. Genau wie in der Formulierung ohne Lift ist dieser Austausch aufgrund des Korollar 2.2 eine äquivalente Umformung. Die KKT-Bedingungen für das Problem lauten

$$\begin{aligned} J^\top(c + c_m) + \bar{N}^\top \lambda + J^\top \mu_1 - \mu_2 &= 0 \\ 0 \leq \mu_1 \perp \begin{aligned} \bar{N}x_u &= b \\ -Jx_u + u - w_1 &\geq 0 \end{aligned} && \geq 0 \\ 0 \leq \mu_2 \perp \begin{aligned} x_u - w_2 &\geq 0. \end{aligned} && \geq 0. \end{aligned}$$

Zusammen mit den Nebenbedingungen an c_m ergeben sich somit für das Problem der oberen Stufe insgesamt folgende Nebenbedingungen:

$$c_m \geq l_{c_m} \tag{2.10a}$$

$$c_m \leq u_{c_m} \tag{2.10b}$$

$$J^\top(c + c_m) + \bar{N}^\top \lambda + J^\top \mu_1 - \mu_2 = 0 \tag{2.10c}$$

$$\bar{N}x_u = b \tag{2.10d}$$

$$0 \leq \mu_1 \perp -Jx_u + u - w_1 \geq 0 \quad (2.10e)$$

$$0 \leq \mu_2 \perp x_u - w_2 \geq 0. \quad (2.10f)$$

Neben den Veränderungen in den Nebenbedingungen durch die Variable $w := \begin{pmatrix} w_1 & w_2 \end{pmatrix}$ wird bei der Modifikation auch ein Strafterm für w der Form $\pi p(w)$ in der Zielfunktion des Problems der oberen Stufe eingeführt. Für die Wahl der Funktion $p(w)$ wurde in [9] die L1-Norm und die quadrierte L2-Norm vorgeschlagen. Bei Verwendung der L1-Norm wurde zusätzlich $w \geq 0$ als Nebenbedingung verwendet, um die numerischen Probleme, welche aufgrund der Nichtdifferenzierbarkeit im Koordinatenursprung auftreten, zu minimieren.

Die Ergebnisse in [9] zeigen, dass mit der L1-Norm im Allgemeinen bessere Resultate erzielt wurden. Versuche haben gezeigt, dass dies auch beim hier behandelten Modell der Fall ist. Aus diesem Grund soll im Folgenden $p(w) = \|w\|_1$ gelten. Der Parameter π dient dazu, dass das Gewicht des Strafterms in der Zielfunktion vergrößert werden kann, falls in einer Lösung $w \neq 0$ gilt. Der daraus resultierende Algorithmus wird in Abschnitt 6.1.2 näher erläutert.

Insgesamt erhält man nun folgendes Problem der oberen Stufe:

$$\begin{aligned} & \underset{\substack{c_m \in \mathbb{R}^m \\ x_u \in \mathbb{R}^{km} \\ \lambda \in \mathbb{R}^{k(n-1)} \\ (\mu_1, \mu_2)^\top, w \in \mathbb{R}^{(k+1)m}}}{\text{minimiere}} & \|u - Jx_u - x_z\|^2 + \pi \|w\|_1 \\ & \text{unter} & \begin{aligned} c_m &\geq l_{c_m} \\ c_m &\leq u_{c_m} \\ J^\top(c + c_m) + \bar{N}^\top \lambda + J^\top \mu_1 - \mu_2 &= 0 \\ \bar{N}x_u - b &= 0 \\ w &\geq 0 \\ 0 \leq \mu_1 \perp u - Jx_u - w_1 &\geq 0 \\ 0 \leq \mu_2 \perp x_u - w_2 &\geq 0. \end{aligned} \end{aligned} \quad (2.11)$$

Analog zur Formulierung ohne Lift sollen nun im Komplementaritätssystem Slackvariablen eingeführt werden, um die numerischen Ergebnisse zu verbessern. Darüber hinaus sollen wieder alle Nebenbedingungen außer den Beschränkungen an c_m sowie w und den Komplementaritätsbedingungen in Matrixschreibweise gefasst werden. Schließlich soll nun auch anders als in [9] die Norm $\|\cdot\|_1$ analog zur Überlegung in [26, Kapitel 17] so ersetzt werden, dass die besprochene Nichtdifferenzierbarkeitsstelle nicht mehr auftritt. Dies wird mittels zusätzlicher Variablen erreicht, die mit v bezeichnet werden sollen. Die Nebenbedingung $w \geq 0$ soll aber weiterhin beibehalten werden, da auf diese Art und Weise bei Tests bessere Ergebnisse erzielt wurden.

Die im Problem der oberen Stufe zu optimierende Variable lautet:

$$\begin{aligned} x_o &= (c_m, x_u, \lambda, \mu_1, \mu_2, w_1, w_2, s_1, s_2, v)^\top, \text{ wobei} \\ x_o &\in \mathbb{R}^{5(k+1)m+k(n-1)} =: \mathbb{R}^{p_2}. \end{aligned}$$

Nun sei $\hat{e} \in \mathbb{R}^{(k+1)m}$ der Vektor, der nur aus Einsen besteht. Das Problem in der Formulierung mit Lift lautet nun

$$\begin{array}{l}
 \text{minimiere} \\
 x_o \in \mathbb{R}^{p_2}
 \end{array}
 \quad \|u - Jx_u - x_z\|^2 + \pi \hat{e}^\top v$$

$$\text{unter} \quad \left(\begin{array}{cccccc}
 J^\top & \bar{N}^\top J^\top & -I & & & \\
 & \bar{N} & & & & \\
 & J & & I & I & \\
 & -I & & & I & I
 \end{array} \right) x_o = \begin{pmatrix} -J^\top c \\ b \\ u \\ 0 \end{pmatrix} \quad (2.12)$$

$$\begin{array}{l}
 c_m \geq l_{c_m} \\
 c_m \leq u_{c_m} \\
 0 \leq w \leq v \\
 0 \leq \mu_1 \perp s_1 \geq 0 \\
 0 \leq \mu_2 \perp s_2 \geq 0.
 \end{array}$$

2.3. Beispiele

Als Abschluss dieses Kapitels soll nun noch einmal die Modellierung anhand zweier konkreter Beispiele verdeutlicht werden. Dabei wird zunächst das Beispiel aus [18] und darüber hinaus noch ein eigenes Beispiel dargestellt.

Beispiel 2.1. vgl. [18]

Dieses Beispiel eines Mautproblems findet auf dem in Abbildung 1 dargestellten Graphen mit 8 Ecken und 11 Kanten statt. Die zugehörigen Kantenbeschränkungen, Kantenkosten und Beschränkungen an c_m sind in Tabelle 1 zusammengefasst. Anhand der Daten aus

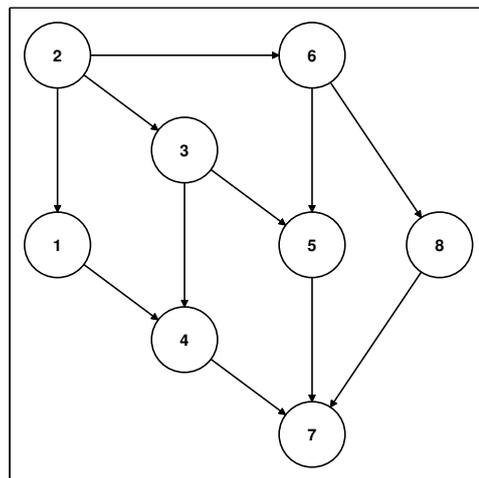


Abbildung 1.: Beispielgraph 2.1

Kante (i, j)	(1,4)	(2,1)	(2,3)	(2,6)	(3,4)	(3,5)	(4,7)	(5,7)	(6,5)	(6,8)	(8,7)
Kantennr. e	1	2	3	4	5	6	7	8	9	10	11
$(u)_e$	15	5	15	15	10	8	10	12	8	7	10
$(x_z)_e$	11	1	10	9	8	5	4	5	4	5	8
$(c)_e$	15	20	23	19	14	11	10	16	14	21	17
$(l_{cm})_e$	0	0	0	0	0	0	0	0	0	0	0
$(u_{cm})_e$	0	0	0	0	0	0	∞	0	0	0	0

Tabelle 1.: Beispieldaten 2.1

Tabelle 1 ergibt sich folgende Inzidenzmatrix \mathcal{N} :

$$\mathcal{N} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

$\hat{\mathcal{N}}$ erhält man nun beispielsweise, indem man die 8. Zeile von \mathcal{N} streicht.

Auf diesem Graphen soll ein Transport der Menge 15 vom Knoten 2 zum Knoten 7 stattfinden, woraus zusammen mit dem Streichen der 8. Zeile aus \mathcal{N}

$$b = \left(0 \quad -15 \quad 0 \quad 0 \quad 0 \quad 0 \quad 15 \right)^\top$$

folgt. Da nur ein Transport stattfindet, gilt hier $\bar{\mathcal{N}} = \hat{\mathcal{N}}$ sowie $J = I$.

In [18] wurde ermittelt, dass

$$c_m = \left(0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \right)^\top$$

zusammen mit

$$\begin{aligned} x &= \lambda x^* + (1 - \lambda)x^{**} && \text{wobei } \lambda \in (0, 1) \\ &\text{sowie } x^* = \left(5 \quad 5 \quad 5 \quad 5 \quad 5 \quad 0 \quad 10 \quad 5 \quad 5 \quad 0 \quad 0 \right) \\ &\text{und } x^{**} = \left(5 \quad 5 \quad 2 \quad 8 \quad 2 \quad 0 \quad 7 \quad 8 \quad 8 \quad 0 \quad 0 \right) \end{aligned}$$

das globale Minimum des Optimierungsproblems (2.8) ist. Zusammen mit $w = 0$, was in einem Optimum stets gilt, trifft dies dann auch auf (2.12) zu.

Beispiel 2.2. Das hier betrachtete Beispiel wird später bei der Untersuchung der Formulierung mit Lift auf Constraint Qualification benötigt und soll schon an dieser Stelle mit vorgestellt werden. Dem Optimierungsproblem liegt der Graph aus Abbildung 2 mit den zugehörigen Daten aus Tabelle 2 zugrunde.

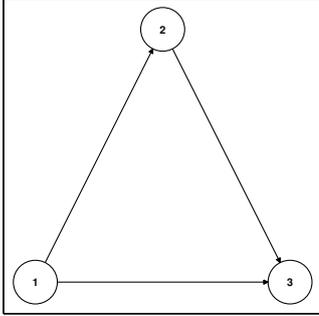


Abbildung 2.: Beispielgraph 2.2

Kante (i,j)	$(1,2)$	$(1,3)$	$(2,3)$
Kantennr. e	1	2	3
$(u)_e$	6	6	6
$(x_z)_e$	0	4	0
$(c)_e$	1	1	1
$(l_{c_m})_e$	0	0	0
$(u_{c_m})_e$	0	∞	0

Tabelle 2.: Beispieldaten 2.2

Die zu dem Graphen gehörige Inzidenzmatrix lautet nun

$$\mathcal{N} = \begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Auch hier wird $\hat{\mathcal{N}}$ aus \mathcal{N} erzeugt, indem die letzte Zeile von \mathcal{N} gestrichen wird. Auf dem Graphen sollen zwei Transporte stattfinden, bei denen jeweils 4 Einheiten vom Knoten 1 zum Knoten 3 fließen sollen. Daraus ergeben sich folgende weitere Vektoren und Matrizen:

$$\bar{\mathcal{N}} = \begin{pmatrix} \hat{\mathcal{N}} & 0 \\ 0 & \hat{\mathcal{N}} \end{pmatrix}, \quad J = \begin{pmatrix} I & I \end{pmatrix} \in \mathbb{R}^{3 \times 6}$$

$$b = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix}, \quad \text{wobei } b^1 = b^2 = (-4, 0, 4)^\top$$

$$x_u = \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}, \quad \text{wobei } x^i \in \mathbb{R}^3 \text{ den Fluss von Ware } i \text{ darstellt.}$$

Der optimale Fluss dieses Beispiels ist durch

$$x_\lambda^* = \lambda \begin{pmatrix} 2 & 2 & 2 & 4 & 0 & 4 \end{pmatrix}^\top + (1-\lambda) \begin{pmatrix} 4 & 0 & 4 & 2 & 2 & 2 \end{pmatrix}^\top$$

mit $\lambda \in [0,1]$ gegeben und wird im Anhang A ausführlich hergeleitet.

3. Untersuchung auf Constraint Qualifications

Ziel dieses Kapitels soll die Untersuchung der Probleme (2.8) und (2.12) auf Constraint Qualifications sein. Zunächst soll jedoch ein Abschnitt zur Vorbereitung dieser Untersuchungen folgen.

3.1. Definitionen und Vorbetrachtungen

Zunächst wird wieder ein allgemeines nichtlineares Optimierungsproblem der Form

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimiere}} & f(x) \\ \text{unter} & g_i(x) \leq 0 \quad \forall i = 1, \dots, m \\ & h_i(x) = 0 \quad \forall i = 1, \dots, p \end{array} \quad (3.1)$$

betrachtet. Für die weiteren Betrachtungen spielen die folgenden Definitionen eine zentrale Rolle.

Definition 3.1. (vgl. [28, S. 358])

Die Vektoren $(a_i)_{i=1, \dots, p} \in \mathbb{R}^n$ heißen linear unabhängig, falls aus $\sum_{i=1}^p \alpha_i a_i = 0$ mit $\alpha_i \in \mathbb{R}$ für $i = 1, \dots, p$ folgt, dass $\alpha_i = 0$ für alle $i = 1, \dots, p$.

Definition 3.2. (vgl. [8, S. 64])

Der Rang einer Matrix M ist die maximale Anzahl an linear unabhängigen Zeilenvektoren oder Spaltenvektoren von M . Dieser wird mit $\text{rank}(M)$ bezeichnet.

Definition 3.3. (vgl. [8, S.64])

Sei $M \in \mathbb{R}^{k_1 \times k_2}$ eine beliebige Matrix. Gilt

- a) $\text{rank}(M) = k_1$, so hat M vollen Zeilenrang.
- b) $\text{rank}(M) = k_2$, so hat M vollen Spaltenrang.

An dieser Stelle sollen nun zwei weit verbreitete Constraint Qualifications vorgestellt werden, die beispielsweise [23] entnommen werden können. Bei den weiteren Betrachtungen gelte dabei $\mathcal{A}(\hat{x}) := \{i \in \{1, \dots, m\} \mid g_i(\hat{x}) = 0\}$.

Definition 3.4. Ein zulässiger Punkt \hat{x} von (3.1) erfüllt die

- a) LICQ (linear independence constraint qualification), wenn die Gradienten

$$\nabla g_i(\hat{x}) \quad (i \in A), \quad \nabla h_i(\hat{x}) \quad (i = 1, \dots, p)$$

linear unabhängig sind.

b) MFCQ (Mangasarian-Fromovitz constraint qualification), wenn die Gradienten $\nabla h_i(\hat{x})$ ($i = 1, \dots, p$) linear unabhängig sind und ein Vektor $d \in \mathbb{R}^n$ existiert, sodass

$$\nabla g_i(\hat{x})^\top d < 0 \quad (i \in A) \quad \text{und} \quad \nabla h_i(\hat{x})^\top d = 0 \quad (i = 1, \dots, p)$$

gilt.

Nun soll das folgende allgemeine MPCC betrachtet werden:

$$\begin{array}{ll} \text{minimiere} & f(x) \\ \text{unter} & \begin{array}{ll} g_i(x) \leq 0 & \forall i = 1, \dots, m \\ h_i(x) = 0 & \forall i = 1, \dots, p \\ G_i(x) \geq 0 & \forall i = 1, \dots, l \\ H_i(x) \geq 0 & \forall i = 1, \dots, l \\ G_i(x)H_i(x) = 0 & \forall i = 1, \dots, l. \end{array} \end{array} \quad (3.2)$$

Aufgrund der Komplementaritätsbedingung erfüllt das Problem (3.2) die MFCQ in keinem zulässigen Punkt (vgl. [6, Lemma 3.1]). In [24, Satz 2.28] wurde gezeigt, dass das Vorliegen der LICQ in einem Punkt x hinreichend für das Vorliegen der MFCQ ist. Da die MFCQ für (3.2) in keinem Punkt erfüllt ist, gilt dies somit auch für die LICQ. Auf dieses Resultat weisen beispielsweise auch Hoheisel, Kanzow und Schwartz in [23] hin, da dies ausschlaggebend für die Entwicklung weiterer Constraint Qualifications speziell für MPCCs war. Wichtig für die Formulierung dieser speziellen Constraint Qualifications sind die Indexmengen

$$\begin{aligned} \mathcal{A}_{+0}(\hat{x}) &= \{i \in \{1, \dots, l\} \mid G_i(\hat{x}) > 0, H_i(\hat{x}) = 0\}, \\ \mathcal{A}_{0+}(\hat{x}) &= \{i \in \{1, \dots, l\} \mid G_i(\hat{x}) = 0, H_i(\hat{x}) > 0\} \quad \text{und} \\ \mathcal{A}_{00}(\hat{x}) &= \{i \in \{1, \dots, l\} \mid G_i(\hat{x}) = 0, H_i(\hat{x}) = 0\}. \end{aligned}$$

Mit diesen Indexmengen wird nun wie in [23] das folgende, vom MPCC (3.2) abgeleitete Problem formuliert:

$$\begin{array}{ll} \text{minimiere} & f(x) \\ \text{unter} & \begin{array}{ll} g_i(x) \leq 0 & \forall i = 1, \dots, m \\ h_i(x) = 0 & \forall i = 1, \dots, p \\ G_i(x) > 0, H_i(x) = 0 & \forall i \in \mathcal{A}_{+0}(\hat{x}) \\ G_i(x) = 0, H_i(x) > 0 & \forall i \in \mathcal{A}_{0+}(\hat{x}) \\ G_i(x) = 0, H_i(x) = 0 & \forall i \in \mathcal{A}_{00}(\hat{x}). \end{array} \end{array} \quad (3.3)$$

Dieses Problem wird tightend nonlinear problem (TNLP) oder auch verschärftes nicht-lineares Problem im Punkt \hat{x} genannt und im folgenden mit $\text{TNLP}(\hat{x})$ bezeichnet.

Mittels dieses Problems werden in [23] folgende MPCC-Constraint Qualifications eingeführt:

Definition 3.5. Ein zulässiger Punkt des MPCC (3.2) erfüllt die

- a) MPCC-LICQ, wenn das zugehörige TNLP(\hat{x}) der LICQ genügt.
 b) MPCC-MFCQ, wenn das zugehörige TNLP(\hat{x}) der MFCQ genügt.

Zur Vereinfachung der Notation werden nun $\mathcal{A}_G(\hat{x}) := \mathcal{A}_{0+}(\hat{x}) \cup \mathcal{A}_{00}(\hat{x})$, $\mathcal{A}_H(\hat{x}) := \mathcal{A}_{+0}(\hat{x}) \cup \mathcal{A}_{00}(\hat{x})$ und $P = \{1, \dots, p\}$ gesetzt sowie die Schreibweisen aus der folgenden Bemerkung verwendet.

Bemerkung 3.1. Es seien $(a_i)_{i \in I}$ eine Menge von Vektoren wobei I eine beliebige endliche Indexmenge ist. Weiterhin sei $\tilde{I} \subseteq I$ mit $\tilde{I} = \{i_1, i_2, \dots, i_{|\tilde{I}|}\}$ eine weitere Indexmenge. Dann gelte folgende Schreibweise :

$$(a_i)^{\tilde{I}} := \begin{pmatrix} | & | & \cdots & | \\ a_{i_1} & a_{i_2} & \cdots & a_{i_{|\tilde{I}|}} \\ | & | & \cdots & | \end{pmatrix}. \quad (3.4)$$

Ist $A \in \mathbb{R}^{q_1 \times q_2}$, so wird $A = \begin{pmatrix} a_1 & \cdots & a_{p_2} \end{pmatrix}$ mit $a_i \in \mathbb{R}^{q_1 \times 1}$ für alle $i \in 1, \dots, q_2$ gesetzt. Analog zu (3.4) gilt dann mit $\tilde{I} \subseteq \{1, \dots, q_2\}$ und $\tilde{I} = \{i_1, i_2, \dots, i_{|\tilde{I}|}\}$ folgende Schreibweise:

$$(A)^{\tilde{I}} := \begin{pmatrix} | & | & \cdots & | \\ a_{i_1} & a_{i_2} & \cdots & a_{i_{|\tilde{I}|}} \\ | & | & \cdots & | \end{pmatrix}.$$

Entscheidend für die weiteren Untersuchungen der Probleme 2.8 und 2.12 auf die Constraint Qualifications MPCC-LICQ und MPCC-MFCQ, ist die nun folgende Bemerkung.

Bemerkung 3.2. Betrachtet man Definition 3.5 und die Struktur des Problems (3.3) so fällt auf, dass sowohl für das Vorliegen der MPCC-LICQ, als auch für das Vorliegen der MPCC-MFCQ die Unabhängigkeit der Gradienten

$$\nabla h_i(\hat{x}) \quad (i \in P), \quad \nabla G_i(\hat{x}) \quad (i \in \mathcal{A}_G(\hat{x})) \quad \text{und} \quad \nabla H_i(\hat{x}) \quad (i \in \mathcal{A}_H(\hat{x}))$$

notwendig ist. Dies ist laut den Definitionen 3.2 und 3.3 dazu äquivalent, dass die Matrix

$$\mathcal{H}(\hat{x}) = \left((\nabla h_i(\hat{x}))^P \quad (\nabla G_i(\hat{x}))^{\mathcal{A}_G(\hat{x})} \quad (\nabla H_i(\hat{x}))^{\mathcal{A}_H(\hat{x})} \right)$$

vollen Spaltenrang hat. Somit ist $\mathcal{H}(\hat{x})$ die transponierte Jacobimatrix der beteiligten Nebenbedingungen.

Bevor die beiden Probleme auf Constraint Qualifications untersucht werden, ist noch das folgende Lemma notwendig.

Lemma 3.1. *Die wie in (2.2) definierte Matrix A hat vollen Zeilenrang.*

Für den Beweis werden die folgenden Resultate aus der Literatur verwendet:

Lemma 3.2. (vgl. [5, Theorem 7-2]) Sei \mathcal{N} die Inzidenzmatrix eines zusammenhängenden Graphen G mit n Knoten. Dann gilt $\text{rank}(\mathcal{N}) = n - 1$.

sowie

Lemma 3.3. (vgl. [10, Lemma 4.5.11.]) Für beliebige Matrizen B_1 und B_2 gilt:

$$\text{rank} \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} = \text{rank}(B_1) + \text{rank}(B_2). \quad (3.5)$$

Nun folgt der Beweis von Lemma 3.1.

Beweis. Wie schon erwähnt, soll der Beweis mit Hilfe von Lemma 3.2 und Lemma 3.3 erfolgen. Bevor Lemma 3.2 angewendet werden kann, muss jedoch sichergestellt werden, dass alle hier verwendeten Graphen zusammenhängend sind. Für die vorgestellten Beispiele 2.1 und 2.2 ist diese Eigenschaft offensichtlich erfüllt. Bei der Erstellung der Beispielgraphen für die umfangreicheren Tests wurde diese Eigenschaft durch den dazu verwendeten Algorithmus B.1 sichergestellt. Damit kann Lemma 3.2 für die hier verwendeten Graphen angewendet werden und man erhält $\text{rank}(\mathcal{N}) = n - 1$. Betrachtet man die Struktur von \mathcal{N} , so stellt sich heraus, dass diese Matrix im Fall eines zusammenhängenden Graphen stets mehr Spalten als Zeilen hat. Dies folgt sofort aus der Überlegung, dass in einem zusammenhängenden Graphen stets mehr Kanten als Knoten existieren müssen. Zusammen mit dem Rang von \mathcal{N} liefert diese Eigenschaft, dass aus \mathcal{N} eine beliebige Zeile herausgelassen werden kann, ohne die Flussbedingung zu verändern. Die auf diese Art und Weise konstruierte Matrix $\hat{\mathcal{N}}$ hat also $n - 1$ Zeilen. Gleichzeitig gilt $\text{rank}(\hat{\mathcal{N}}) = n - 1$, woraus folgt, dass $\hat{\mathcal{N}}$ vollen Zeilenrang hat. Nach 2.2 gilt

$$\bar{\mathcal{N}} = \begin{pmatrix} \hat{\mathcal{N}} & & 0 \\ & \ddots & \\ 0 & & \hat{\mathcal{N}} \end{pmatrix}.$$

Wendet man nun Lemma 3.3 $(k - 1)$ -mal an, so erhält man, dass $\bar{\mathcal{N}}$ vollen Zeilenrang aufweist, da $\hat{\mathcal{N}}$ vollen Zeilenrang hat. \square

3.2. Formulierung ohne Lift

Zur Vereinfachung der weiteren Notation werden nun weitere Indexmengen definiert.

Bemerkung 3.3. Sei $\hat{x}_o = (\hat{c}_m, \hat{x}_u, \hat{\lambda}, \hat{\mu}_1, \hat{\mu}_2, \hat{s}_1)^\top$ ein zulässiger Punkt von (2.8). Dann werden folgende Indexmengen definiert:

$$\begin{aligned} \mathcal{A}_{\hat{\mu}_1} &= \{i \in \{1, \dots, m\} \mid (\hat{\mu}_1)_i = 0\} \\ \mathcal{A}_{\hat{\mu}_2} &= \{i \in \{1, \dots, km\} \mid (\hat{\mu}_2)_i = 0\} \\ \mathcal{A}_{\hat{s}_1} &= \{i \in \{1, \dots, m\} \mid (\hat{s}_1)_i = 0\} \\ \mathcal{A}_{\hat{x}_u} &= \{i \in \{1, \dots, km\} \mid (\hat{x}_u)_i = 0\}. \end{aligned}$$

Zur nun folgenden Untersuchung auf Constraint Qualifications soll die Überlegung aus Bemerkung 3.2 angewendet werden, nach der das Problem (2.8) nur dann die MPCC-LICQ oder die MPCC-MFCQ erfüllen kann, wenn die Matrix

$$\begin{aligned} \mathcal{H}_1(\hat{x}_o) &= \begin{pmatrix} (\nabla h_i(\hat{x}))^P & (\nabla G_i(\hat{x}))^{\mathcal{A}_G(\hat{x})} & (\nabla H_i(\hat{x}))^{\mathcal{A}_H(\hat{x})} \\ J & \bar{\mathcal{N}}^\top & J^\top & & & & I^{\mathcal{A}_{\hat{x}_u}} \\ \bar{\mathcal{N}} & & & & & & & \\ J & & & I^{\mathcal{A}_{\hat{\mu}_1}} & & & & \\ -I & & & & I^{\mathcal{A}_{\hat{\mu}_2}} & & & \\ & & I & & & & I^{\mathcal{A}_{\hat{s}_1}} & \end{pmatrix} \begin{pmatrix} c_m \\ x_u \\ \lambda \\ \mu_1 \\ \mu_2 \\ s_1 \end{pmatrix} \\ &= \begin{pmatrix} J & \bar{\mathcal{N}}^\top & J^\top & & & & I^{\mathcal{A}_{\hat{x}_u}} \\ \bar{\mathcal{N}} & & & & & & & \\ J & & & I^{\mathcal{A}_{\hat{\mu}_1}} & & & & \\ -I & & & & I^{\mathcal{A}_{\hat{\mu}_2}} & & & \\ & & I & & & & I^{\mathcal{A}_{\hat{s}_1}} & \end{pmatrix} \begin{pmatrix} c_m \\ x_u \\ \lambda \\ \mu_1 \\ \mu_2 \\ s_1 \end{pmatrix} \end{aligned}$$

vollen Spaltenrang hat. Es gilt aber folgendes Lemma:

Lemma 3.4. *Sei \hat{x}_o ein zulässiger Punkt von (2.8). Existiert eine Ecke $\tilde{g} \in N \setminus \{q\}$, bei der für alle Kanten $e \in A_{\tilde{g}} := A_1 \dot{\cup} A_2$ mit $A_1 = \{e \in A \mid e = (\tilde{g}, v), v \in N\}$ sowie $A_2 = \{e \in A \mid e = (v, \tilde{g}), v \in N\}$ genau eine der beiden Bedingungen*

(i) $\hat{x}_e^l = 0$ für alle $l \in K$ **oder**

(ii) $\sum_{l=1}^k \hat{x}_e^l = u_e$.

gilt, dann hat die Matrix $\mathcal{H}_1(\hat{x}_o)$ keinen vollen Spaltenrang und somit erfüllt das Problem (2.8) weder die MPCC-LICQ noch die MPCC-MFCQ.

Beweis. Der Beweis erfolgt in zwei Abschnitten. Im ersten Teil wird gezeigt, dass das Gleichungssystem

$$\bar{\mathcal{N}}^\top y_1 + J^\top I^{\mathcal{A}_{\hat{s}_1}} y_2 + I^{\mathcal{A}_{\hat{x}_u}} y_3 = 0 \quad (3.6)$$

eine nichttriviale Lösung hat. Dies wird anschließend verwendet, um zu zeigen, dass $\mathcal{H}_1(\hat{x}_o)$ keinen vollen Spaltenrang hat.

a) Sei nun \hat{x}_o ein zulässiger Punkt von (2.8) und $\tilde{g} \in N \setminus \{q\}$ eine Ecke sodass die Bedingung aus dem Lemma erfüllt ist. Im Weiteren sei mit \mathbf{e}_i der i -te Einheitsvektor bezeichnet. Außerdem werden die Mengen

$$\begin{aligned} C_1 &= \{e \in A_{\tilde{g}} \mid e \text{ erfüllt Eigenschaft (i)}\} \text{ und} \\ C_2 &= \{e \in A_{\tilde{g}} \mid e \text{ erfüllt Eigenschaft (ii)}\}. \end{aligned}$$

benötigt. Nun werden die y_i mit $i = 1, 2, 3$ wie folgt gesetzt:

- y_1
Da $\tilde{g} \in N \setminus \{q\}$ gibt es eine zum Knoten \tilde{g} korrespondierende Zeile in $\bar{\mathcal{N}}$ die

mit c bezeichnet sei. Im Hinblick auf die spezielle Struktur von $\bar{\mathcal{N}}$, kann y_1 wie folgt gesetzt werden:

$$y_1 = \left(y_1^1 \quad \cdots \quad y_1^k \right)^\top \quad \text{wobei } y_1^l = \mathbf{e}_c \text{ f\"ur } l \in K.$$

- y_2

Nun soll mit \hat{e} die Nummer der zur Kante e korrespondierenden Spalte von $J^{\mathcal{A}_{s_1}}$ bezeichnet werden. y_2 wird nun wie folgt gesetzt:

$$(y_2)_j = \begin{cases} 1, & \text{falls } \exists e \in A_1 \cap C_2 : j = \hat{e} \\ -1, & \text{falls } \exists e \in A_2 \cap C_2 : j = \hat{e} \\ 0, & \text{sonst.} \end{cases}$$

- y_3

Auch hier wird analog zu y_2 mit \hat{e}_l die Nummer der zur Kante e des Transports der Ware l korrespondierenden Spalte von $I^{\mathcal{A}_{x_u}}$ bezeichnet. y_3 wird nun wie folgt gesetzt:

$$(y_3)_j = \begin{cases} 1, & \text{falls } \exists l \in K, e \in A_1 \cap C_1 : j = \hat{e}_l \\ -1, & \text{falls } \exists l \in K, e \in A_2 \cap C_1 : j = \hat{e}_l \\ 0, & \text{sonst.} \end{cases}$$

Das Gleichungssystem (3.6) wird nun umgeformt:

$$\begin{aligned} 0 &= \bar{\mathcal{N}}^\top y_1 + J^{\mathcal{A}_{s_1}} y_2 + I^{\mathcal{A}_{x_u}} y_3 \\ &= \begin{pmatrix} \hat{\mathcal{N}}^\top & & 0 \\ & \ddots & \\ 0 & & \hat{\mathcal{N}}^\top \end{pmatrix} \begin{pmatrix} y_1^1 \\ \vdots \\ y_1^k \end{pmatrix} + \begin{pmatrix} I \\ \vdots \\ I \end{pmatrix}^{\mathcal{A}_{s_1}} y_2 + I^{\mathcal{A}_{x_u}} y_3 \\ - \begin{pmatrix} \hat{\mathcal{N}}_{\tilde{g}}^\top \\ \vdots \\ \hat{\mathcal{N}}_{\tilde{g}}^\top \end{pmatrix} &= \begin{pmatrix} I \\ \vdots \\ I \end{pmatrix}^{\mathcal{A}_{s_1}} y_2 + I^{\mathcal{A}_{x_u}} y_3 =: M_1 y_2 + M_2 y_3 \end{aligned} \quad (3.7)$$

wobei $\hat{\mathcal{N}}_{\tilde{g}}$ die zum Knoten \tilde{g} geh\"orende Zeile von $\hat{\mathcal{N}}$ ist. Wichtig f\"ur die weiteren Betrachtungen ist, dass aufgrund der Struktur von M_1 jede zu einer Kante e korrespondierende Spalte von M_1 Einfluss auf genau die Zeilen der Gleichung hat, die f\"ur alle Waren l die Kante e repr\"asentieren. F\"ur die Matrix M_2 gilt dagegen, dass eine zu \hat{x}_e^l korrespondierende Spalte Einfluss auf genau die Zeile der Gleichung hat, die \hat{x}_e^l repr\"asentiert. In der nun folgenden Fallunterscheidung werden alle Kanten $e \in A$ durchlaufen und dabei bei jeder Kante die G\"ultigkeit der Gleichung (3.7) auf allen zu dieser Kante korrespondierenden Zeilen gezeigt.

Fall 1: $e \notin A_{\bar{g}}$

Auf der linken Seite von (3.7) stehen auf den zu e korrespondierenden Zeilen jeweils Nullen, was sich direkt daraus ergibt, dass in einer zum Knoten gehörenden Zeile nur bei den zu diesem Knoten inzidenten Kanten Nicht-nulleinträge auftreten.

Auf der rechten Seite wird deutlich, dass die zu e korrespondierenden Einträge von y_2 und y_3 gleich Null gesetzt wurden. Somit treten auch auf der rechten Seite auf den betreffenden Zeilen nur Nullen auf.

Fall 2: $e \in A_1$

Die linke Seite von (3.7) ergibt in diesem Fall 1, was man direkt aus der Definition von $\hat{\mathcal{N}}$ und der Multiplikation mit -1 erhält. Laut den Voraussetzung muss nun für e genau einer dieser beiden Fälle zutreffen.

Fall 2a: $e \in A_1 \cap C_1$

Da $e \in C_1$ gilt nach den Voraussetzungen, dass $x_e^l = 0$ für alle $l \in K$ und somit gibt es für alle Waren l eine zu e korrespondierende Spalte in M_2 und somit auch jeweils eine zu der Spalte korrespondierende Komponente in y_3 . Mit der beschriebenen Setzung von y_3 erhält man, dass genau auf jeder zu e korrespondierenden Zeile 1 steht.

Fall 2b: $e \in A_1 \cap C_2$

Aus $e \in C_2$ folgt wegen den Voraussetzungen und $s_1 = u - \sum_{l=1}^k \hat{x}^l$, dass die zu e gehörende Spalte in M_1 enthalten sein muss. Durch die Wahl von y_2 ergibt sich, dass diese Spalte mit 1 multipliziert wird, woraus folgt, dass auch in diesem Fall in jeder zu e korrespondierenden Zeile der rechten Seite 1 steht.

Da stets nur einer dieser beiden Fälle eintritt, gilt in allen zu e gehörenden Zeilen von (3.7) Gleichheit.

Fall 3: $e \in A_2$

Dieser Fall soll analog zu Fall 2 behandelt werden. Auf der linken Seite steht nun auf jeder betreffenden Zeile -1 , da man aus der Definition von $\hat{\mathcal{N}}$ jeweils eine 1 erhält die noch mit -1 multipliziert wird. Nun gilt für e wieder genau einer der folgenden Fälle.

Fall 3a: $e \in A_2 \cap C_1$

Hier gilt, dass auf jeder der Zeilen eine -1 auftritt, was sich durch die andere Setzung von y_3 im Gegensatz zum Fall 2a ergibt. Die übrigen Überlegungen erfolgen analog.

Fall 3b: $e \in A_2 \cap C_2$

Auch hier erhält man in jeder zu e korrespondierenden Zeilen der rechten Seite eine -1 , was wieder durch zum Fall 2b analoge Überlegungen, aber das entgegengesetzte Vorzeichen bei der Setzung von y_2 , ergibt.

□

Beim Betrachten der Bedingungen an das Problem (2.8) und die Lösung \hat{x}_o fällt auf, dass nur eine der beiden Bedingungen gelten darf. Das gleichzeitige Auftreten beider Bedingungen ändert an der Gültigkeit dieses Lemmas nichts, allerdings würde dies bedeuten, dass über eine Kante e kein Transport stattfindet und gleichzeitig die obere Kapazitätsgrenze erreicht ist. Somit hat diese Kante eine Kapazität $u_e = 0$. Derartige Kanten werden in der Modellierung einer konkreten Problemstellung nicht auftreten.

Gleichzeitig wurde nur bewiesen, dass das Kriterium in 3.4 nur hinreichend dafür ist, dass $\mathcal{H}_1(\hat{x}_o)$ keinen vollen Zeilenrang hat. Gleichwohl ist schon allein die Sicherstellung, dass dieses Kriterium für ein Problem nicht erfüllt ist, algorithmisch sehr schwierig, insbesondere da man für verlässliche Resultate diese Bedingung für alle zulässigen \hat{x}_o sicher verhindern müsste.

Nun soll noch gezeigt werden, dass die Bedingung aus Lemma 3.4 auch an einem Beispiel erfüllt ist. Dazu wird Beispiel 2.1 verwendet. Betrachtet man die Lösung so erhält man, dass über alle an Knoten 8 anliegenden Kanten kein Transport stattfindet. Somit ist die Voraussetzung aus Lemma 3.4 erfüllt und man erhält, dass das Beispiel 2.1 in der Formulierung (2.8) weder die MPCC-LICQ noch die MPCC-MFCQ erfüllt.

3.3. Formulierung mit Lift

Die Intension zur Einführung des Lifts in [9] war es, unter gewissen Annahmen an das Problem der unteren Stufe zu zeigen, dass das Problem der oberen Stufe stets die MPCC-LICQ erfüllt. Zunächst sollen wieder einige Indexmengen definiert werden:

Bemerkung 3.4. Sei $\hat{x}_o = (\hat{c}_m, \hat{x}_u, \hat{\lambda}, \hat{\mu}_1, \hat{\mu}_2, \hat{w}_1, \hat{w}_2, \hat{s}_1, \hat{s}_2)^\top$ ein zulässiger Punkt von (2.8). Dann soll gelten:

$$\begin{aligned} \mathcal{A}_{\hat{\mu}_1} &:= \{i \in \{1, \dots, m\} \mid (\hat{\mu}_1)_i = 0\} \\ \mathcal{A}_{\hat{\mu}_2} &:= \{i \in \{1, \dots, km\} \mid (\hat{\mu}_2)_i = 0\} \\ \mathcal{A}_{\hat{s}_1} &:= \{i \in \{1, \dots, m\} \mid (\hat{s}_1)_i = 0\} \\ \mathcal{A}_{\hat{s}_2} &:= \{i \in \{1, \dots, km\} \mid (\hat{s}_2)_i = 0\}. \end{aligned}$$

Laut [9, Theorem 2.1.] ist es nun insbesondere notwendig, dass das Problem (3.9), welches

$$\begin{aligned} &\underset{x_u \in \mathbb{R}^{km}}{\text{minimiere}} && \left(J^\top(c + c_m) \right)^\top x_u \\ &\text{unter} && \begin{aligned} \bar{N}x_u - b &= 0 \\ -Jx_u + u &\geq w_1 \\ x_u &\geq w_2. \end{aligned} \end{aligned} \tag{3.9}$$

lautete, die LICQ in der Lösung $x_o^{*\top} = (c_m^*, x_u^*, \lambda^*, \mu_1^*, \mu_2^*, w_1^*, w_2^*, s_1^*, s_2^*)$ erfüllt. Nach der Konstruktion des Lifts muss jedoch in einer Lösung $w_1^* = 0$ und $w_2^* = 0$ gelten. Dies sorgt

- Gleichung (3.12b)

$$\begin{aligned}
 0 &= \bar{\mathcal{N}} y_1 \\
 \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} &= \begin{pmatrix} \hat{\mathcal{N}} & & \\ & \ddots & \\ & & \hat{\mathcal{N}} \end{pmatrix} \begin{pmatrix} y_1^1 \\ \vdots \\ y_1^k \end{pmatrix} \\
 \begin{pmatrix} 0 \\ 0 \end{pmatrix} &= \begin{pmatrix} \hat{\mathcal{N}} y_1^i \\ \hat{\mathcal{N}} y_1^j \end{pmatrix} \\
 \begin{pmatrix} 0 \\ 0 \end{pmatrix} &= \begin{pmatrix} \sum_{e \in W_1} \hat{\mathcal{N}}^{\{e\}} (y_1^i)_e + \sum_{e \in W_2} \hat{\mathcal{N}}^{\{e\}} (y_1^i)_e \\ \sum_{e \in W_1} \hat{\mathcal{N}}^{\{e\}} (y_1^j)_e + \sum_{e \in W_2} \hat{\mathcal{N}}^{\{e\}} (y_1^j)_e \end{pmatrix} \\
 \begin{pmatrix} 0 \\ 0 \end{pmatrix} &= \begin{pmatrix} \sum_{e \in W_1} \hat{\mathcal{N}}^{\{e\}} - \sum_{e \in W_2} \hat{\mathcal{N}}^{\{e\}} \\ - \sum_{e \in W_1} \hat{\mathcal{N}}^{\{e\}} + \sum_{e \in W_2} \hat{\mathcal{N}}^{\{e\}} \end{pmatrix}
 \end{aligned}$$

Hierbei sei $\hat{\mathcal{N}}^{\{e\}}$ analog zur schon eingeführten Schreibweise die e -te Spalte von $\hat{\mathcal{N}}$. Da W_1 und W_2 Wege vom Knoten i zum Knoten j sind, gilt

$$\sum_{e \in W_1} \hat{\mathcal{N}}^{\{e\}} = \sum_{e \in W_2} \hat{\mathcal{N}}^{\{e\}} = \mathbf{e}_j - \mathbf{e}_i,$$

wobei mit \mathbf{e}_p wieder der p -te Einheitsvektor bezeichnet wird. Somit ist auch Gleichung (3.12b) erfüllt.

- Gleichung (3.12c) Es gilt

$$\begin{aligned}
 I y_1 &= I^{\mathcal{A}_{\hat{\mu}_2}}(y_1)_{\mathcal{A}_{\hat{\mu}_2}} \\
 y_1 &= q \quad \text{mit } (q)_e = \begin{cases} (y_1)_e, & \text{falls } (\hat{\mu}_2)_e = 0 \\ 0, & \text{sonst.} \end{cases}
 \end{aligned}$$

Da \hat{x}_o eine Lösung von (2.12) ist, muss es auch zulässig sein und somit gilt insbesondere $\hat{s}_2 = \hat{x}_u - \hat{w}_2$ und $0 \leq \hat{\mu}_2 \perp \hat{s}_2 \geq 0$. Nach der Konstruktion des Lifts liegt in einer Lösung $\hat{w}_2 = 0$ vor. Damit kann im Weiteren $0 \leq \hat{\mu}_2 \perp \hat{x}_u \geq 0$ vorausgesetzt werden. Die Gültigkeit von $y_1 = q$ wird nun mit einer Fallunterscheidung, die für jede Komponente $e \in \mathbb{R}^{km}$ durchgeführt wird, gezeigt.

Fall 1: $(\hat{\mu}_2)_e = 0$

In diesem Fall gilt die Gleichung trivialerweise.

Fall 2: $(\hat{\mu}_2)_e > 0$

In diesem Fall steht auf der rechten Seite der Gleichung eine 0, was aus der Definition von q folgt.

Betrachtet man die Setzung von y_1 , so wird deutlich, dass $(y_1)_e > 0$ nur dann gelten kann, falls $(\hat{x}_u)_e > 0$. Mit $(\hat{\mu}_2)_e > 0$ und der Komplementarität von $\hat{\mu}_2$ und \hat{x}_u erhält man jedoch $(\hat{x}_u)_e = 0$ und somit auch $(y_1)_e = 0$.

Damit gilt auch die Gleichung (3.12c).

Insgesamt erhält man also, dass die dargestellte Setzung des $y = (y_1 \ \cdots \ y_8)^\top$ die Gleichung $0 = \mathcal{H}_2(\hat{x}_o)y$ erfüllt und gleichzeitig y_1 und y_6 nichttriviale Einträge haben. Somit wurde gezeigt, dass $\mathcal{H}_2(\hat{x}_o)$ keinen vollen Spaltenrang hat, woraus folgt, dass das Problem (2.12) im Punkt \hat{x}_o weder die MPCC-LICQ noch die MPCC-MFCQ erfüllen kann. \square

An einem Beispiel soll nun noch verdeutlicht werden, dass die im Lemma 3.5 beschriebenen Anforderungen an eine Lösung \hat{x}_o auftreten können. Dazu soll nun Beispiel 2.2 mit dem optimalen Fluss $x_{\frac{1}{2}}$ betrachtet werden, welcher im Anhang A beschrieben wurde. Man erhält also für die einzelnen Waren die Flüsse

$$x^1 = x^2 = (3 \ 1 \ 3)^\top.$$

Es wird deutlich, dass der Transport beider Waren jeweils über die Wege $W_1 = 1, 3$ und $W_2 = 1, 2, 3$ erfolgt, die jeweils vom ersten zum dritten Knoten führen. Somit sind hier die Voraussetzungen von Lemma 3.5 erfüllt und damit folgt, dass in dieser Lösung das Problem (2.12) weder die MPCC-LICQ noch die MPCC-MFCQ erfüllt.

4. Regularisierungsverfahren

Im nun folgenden Abschnitt sollen die in [23] vorgestellten Regularisierungsverfahren für MPCCs kurz zusammengefasst werden, da diese bei der numerischen Lösung der MPCCs (2.8) und (2.12) angewendet wurden. Die Grundidee eines Regularisierungsverfahrens ist dabei die zulässige Menge durch einen Parameter $t > 0$ zu vergrößern. Um eine Lösung des ursprünglichen Problems zu erhalten, wird dann eine Folge von Optimierungsproblemen mit $t \rightarrow 0$ gelöst.

Der Unterschied zwischen diesen verschiedenen Regularisierungsverfahren betrifft dabei die Art und Weise mit der die Modifikation des zulässigen Bereiches erfolgt.

Im Weiteren wird ein allgemeines MPCC der Form

$$\begin{array}{ll}
 \underset{x \in \mathbb{R}^n}{\text{minimiere}} & f(x) \\
 \text{unter} & g_i(x) \leq 0 \quad \forall i = 1, \dots, m \\
 & h_i(x) = 0 \quad \forall i = 1, \dots, p \\
 & G_i(x) \geq 0 \quad \forall i = 1, \dots, l \\
 & H_i(x) \geq 0 \quad \forall i = 1, \dots, l \\
 & G_i(x)H_i(x) = 0 \quad \forall i = 1, \dots, l
 \end{array} \tag{4.1}$$

betrachtet. Zur numerischen Lösung dieses MPCCs ohne Regularisierung wurde in [20] die Form

$$\begin{array}{ll}
 \underset{x \in \mathbb{R}^n}{\text{minimiere}} & f(x) \\
 \text{unter} & g(x) \leq 0 \\
 & h(x) = 0 \\
 & G(x) \geq 0 \\
 & H(x) \geq 0 \\
 & G(x)^\top H(x) \leq 0
 \end{array} \tag{4.2}$$

sowie das Einführen von Slackvariablen im Komplementaritätssystem vorgeschlagen. Zum Vergleich mit den Regularisierungen sowie um einen Vergleich mit den Ergebnissen aus [9] herzustellen, sollen die Probleme auch ohne Regularisierungsverfahren in der obigen Formulierung gelöst werden, wobei das besprochene Einführen von Slackvariablen im Komplementaritätssystem in der Modellierung bereits vorgenommen wurde.

Der Fokus soll im Weiteren jedoch auf dem Test der in [23] thematisierten Regularisierungsverfahren liegen, die das Problem (4.1) in folgende Optimierungsprobleme überführen:

- Globale Relaxierung nach Scholtes (S), vorgestellt in [21]

$$\begin{array}{ll}
\text{minimiere} & f(x) \\
\text{unter} & \begin{array}{ll}
g_i(x) \leq 0 & \forall i = 1, \dots, m \\
h_i(x) = 0 & \forall i = 1, \dots, p \\
G_i(x) \geq 0 & \forall i = 1, \dots, l \\
H_i(x) \geq 0 & \forall i = 1, \dots, l \\
G_i(x)H_i(x) \leq t & \forall i = 1, \dots, l
\end{array}
\end{array}$$

- Relaxierung nach Lin und Fukushima (LF), vorgestellt in [7]

$$\begin{array}{ll}
\text{minimiere} & f(x) \\
\text{unter} & \begin{array}{ll}
g_i(x) \leq 0 & \forall i = 1, \dots, m \\
h_i(x) = 0 & \forall i = 1, \dots, p \\
G_i(x)H_i(x) - t^2 \leq 0 & \forall i = 1, \dots, l \\
(G_i(x) + t)(H_i(x) + t) - t^2 \geq 0 & \forall i = 1, \dots, l
\end{array}
\end{array}$$

- Relaxierung nach Kadrani et al. (KDB), vorgestellt in [1]

$$\begin{array}{ll}
\text{minimiere} & f(x) \\
\text{unter} & \begin{array}{ll}
g_i(x) \leq 0 & \forall i = 1, \dots, m \\
h_i(x) = 0 & \forall i = 1, \dots, p \\
G_i(x) \geq -t & \forall i = 1, \dots, l \\
H_i(x) \geq -t & \forall i = 1, \dots, l \\
(G_i(x) - t)(H_i(x) - t) \leq 0 & \forall i = 1, \dots, l
\end{array}
\end{array}$$

- Lokale Relaxierung nach Steffensen und Ulbrich (SU), vorgestellt in [25]

$$\begin{array}{ll}
\text{minimiere} & f(x) \\
\text{unter} & \begin{array}{ll}
g_i(x) \leq 0 & \forall i = 1, \dots, m \\
h_i(x) = 0 & \forall i = 1, \dots, p \\
G_i(x) \geq 0 & \forall i = 1, \dots, l \\
H_i(x) \geq 0 & \forall i = 1, \dots, l \\
\Phi_1(G_i(x), H_i(x); t) \leq 0 & \forall i = 1, \dots, l
\end{array}
\end{array}$$

wobei $\Phi_1 : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ $\Phi_1(x_1, x_2; t) = x_1 + x_2 - \phi_1(x_1 - x_2; t)$, mit

$$\phi_1(\cdot; t) : \mathbb{R} \rightarrow \mathbb{R}, \quad \phi_1(a; t) := \begin{cases} |a|, & \text{falls } |a| \geq t \\ t\theta\left(\frac{a}{t}\right), & \text{sonst.} \end{cases}$$

Die Funktion $\theta : [-1, 1] \rightarrow \mathbb{R}$ muss dabei folgende Eigenschaften erfüllen:

- θ ist auf $[-1, 1]$ zweimal stetig differenzierbar;

- $\theta(-1) = \theta(1) = 1$;
- $\theta'(-1) = -1$, sowie $\theta'(1) = 1$;
- $\theta''(-1) = \theta''(1) = 0$;
- $\theta''(z) > 0$ für $z \in (-1, 1)$.

In [25] wurde die Funktion $\theta(z) = \frac{2}{\pi} \sin\left(z\frac{\pi}{2} + \frac{3\pi}{2}\right) + 1$ vorgeschlagen, die für die numerischen Versuche auch Anwendung fand.

- Relaxierung nach Kanzow und Schwartz (KS), vorgestellt in [22]

$$\begin{array}{ll}
 \underset{x \in \mathbb{R}^n}{\text{minimiere}} & f(x) \\
 \text{unter} & g_i(x) \leq 0 \quad \forall i = 1, \dots, m \\
 & h_i(x) = 0 \quad \forall i = 1, \dots, p \\
 & G_i(x) \geq 0 \quad \forall i = 1, \dots, l \\
 & H_i(x) \geq 0 \quad \forall i = 1, \dots, l \\
 & \Phi_2(G_i(x), H_i(x); t) \leq 0 \quad \forall i = 1, \dots, l
 \end{array}$$

wobei $\Phi_2 : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ $\Phi_2(x_1, x_2; t) = \phi_2(x_1 - t, x_2 - t)$ und

$$\phi_2 : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \phi_2(a, b) := \begin{cases} ab, & \text{falls } a + b \geq 0 \\ -\frac{1}{2}(a^2 + b^2), & \text{sonst.} \end{cases}$$

Bevor der numerische Vergleich dieser Regularisierungsverfahren erfolgen kann, müssen noch passende Beispielpunkte generiert werden. Auf welche Art und Weise dies umgesetzt wurde, soll im nächsten Kapitel dargestellt werden.

5. Beispielprobleme

Der Test der vorgestellten numerischen Verfahren kann nur anhand von Beispielproblemen erfolgen. In [23] und [9] wurden die numerischen Untersuchungen jeweils an der MacMPEC Problemsammlung [17] durchgeführt. Diese Sammlung umfasst verschiedene Beispiele für MPCCs, insbesondere auch einzelne Transportprobleme, wie die Probleme *tollmpec* und *tollmpec1*. Bei diesen Problemen wird die Zielfunktion im Problem der oberen Stufe jedoch so formuliert, dass in dem zugrundeliegenden Netzwerk Stau minimiert werden soll beziehungsweise die Gesamteinnahmen durch die Maut zu maximieren sind. Da bei der hier vorgestellten Modellierung im Gegensatz dazu die Restkapazitäten auf dem Netzwerk an die vorher vorgegebenen Restkapazitäten x_z angenähert werden sollen, sind diese Beispiele für die hier notwendigen Untersuchungen nicht geeignet. Dadurch wird es notwendig eine passende Aufgabenkollektion zu erstellen.

Die Erstellung der Beispielprobleme erfolgt mittels des Algorithmus B.1, der im Anhang B dargestellt ist. Als Eingabegrößen für diesen Algorithmus sind eine Anzahl an Transporten k und Knotenzahl \hat{n} notwendig. Dabei ist zu beachten, dass die Knotenzahl des Graphen, der zu dem erzeugten Optimierungsproblem gehört, tatsächlich in der Menge $\{\lceil 0.7 \hat{n} \rceil, \dots, \hat{n}\}$ liegt. Darüber hinaus ist ein Parameter $p \in (0, 2)$ notwendig, über den die Anzahl der zu setzenden Kanten gesteuert wird. Dieser Parameter wurde so gewählt, dass circa 60 % der Kanten im Graphen gesetzt werden. Damit weist der Graph ausreichend Verzweigungen auf und gleichzeitig werden nicht zu viele Kanten gesetzt, was dazu führen würde, dass die Waren zu kurze Wege hätten. In der nachfolgenden Tabelle ist dargestellt, welche Anzahl an Problemen mit welchen Parametern erzeugt wurden. Dabei ist zu beachten, dass bei der angegebenen Anzahl nur diejenigen Probleme gezählt wurden, welche von Algorithmus B.1 als gültige Aufgaben ausgegeben wurden.

Anzahl der Probleme	\hat{n}	k	p	durchschnittlicher Anteil der gesetzten Kanten
20	6	3	0,9	62,2 %
20	8	5	1	57,7 %

Tabelle 3.: Beispielprobleme

6. Numerische Betrachtungen

6.1. Grundlegende verwendete Algorithmen

Zu Beginn dieses Abschnitts sollen die grundlegenden Routinen dargestellt werden, die beim Lösen der Beispielprobleme verwendet wurden. Die beiden wichtigsten sind dabei die algorithmische Umsetzung der Regularisierung sowie die Bestimmung der Lösung eines Problems, welches nach dem Ansatz von Hatz et al. modifiziert wurde.

6.1.1. Algorithmen für die Regularisierungen

Zunächst soll das algorithmische Vorgehen bei der Lösung von regularisierten Problemen beschrieben werden. Wie schon in Kapitel 4 dargestellt, ist die Grundidee der Regularisierungen eine Folge von Problemen mit einem Regularisierungsparameter $t \searrow 0$ zu lösen. Die darauf aufbauenden Algorithmen wurden in [23] beschrieben und sind im Anhang C dargestellt. Wichtig ist dabei, dass für alle Regularisierungen, außer der von Kandrani et al. ein verbesserter Algorithmus verwendet werden kann. Dieser nutzt die Eigenschaft, dass für diese Regularisierungen das zulässige Gebiet $X(t_k)$ im Schritt t_k eine Teilmenge des Gebietes $X(t_{k-1})$ ist und somit eine im Schritt $k-1$ gefundene Lösung x_{k-1} auch im Schritt t_k optimal ist, falls $x_{k-1} \in X(t_k)$ gilt.

Als Lösungsroutine wird die MATLAB-Funktion *fmincon* mit dem SQP-Verfahren verwendet. Dabei handelt es sich um ein Verfahren, bei dem in jeder Iteration ein quadratisches Teilproblem gelöst wird. Mittels den dabei erhaltenen Lösungen sowie den zugehörigen Multiplikatoren wird dann schrittweise ein KKT-Punkt des ursprünglichen Problems berechnet (vgl. [19, Abschnitt 19]). Die guten Eigenschaften dieses Verfahrens für die Lösung von MPCCs ohne Regularisierungsverfahren wurden von Flechter et al. in [20] diskutiert.

In MATLAB sind neben dem SQP-Verfahren das Active-Set, Interior-Point sowie das Trust-Region-Reflektive Verfahren implementiert. Ein Vergleich dieser Lösungsverfahren hat ergeben, dass auch für die Lösung der MPCCs mit einer Regularisierung das SQP-Verfahren die besten Resultate liefert. Aus diesem Grund wird bei der Implementierung die Funktion *fmincon* mit dem SQP-Verfahren verwendet.

6.1.2. Umsetzung des Lifts

Nun soll beschrieben werden, wie das Lifting-Verfahren umgesetzt wurde. Die dazu verwendete Routine ist in Algorithmus 1 dargestellt.

Der erste entscheidende Abschnitt in diesem Algorithmus ist, wie auch in [9] dargestellt, das Finden eines Parameters w , sodass die durch dieses w veränderten Nebenbe-

Algorithmus 1 Algorithmus zum Lösen der gelifteten Probleme**Eingabe:** Start-Liftingparameter w und $\pi, \hat{\varepsilon}, \pi_{\max}$ **Ausgabe:** zugehöriger Zielfunktionswert $fval$

```

1: Teste ob das Problem mit Parametern  $w$  und  $\pi$  eine Lösung  $(x_o)_0$  hat
2: while  $\nexists$  Lösung  $(x_o)_0$  des Problems mit Parametern  $w$  und  $\pi$  do
3:    $w := 0,1 \cdot w$ 
4:   Teste ob das Problem mit Parametern  $w$  und  $\pi$  eine Lösung  $(x_o)_0$  hat
5: end while
6: Rufe Alg. C.1 bzw. C.2 für das Problem mit den Parametern  $w, \pi$  sowie Startlösung
    $(x_o)_0$  auf und setze  $x_o$  und  $fval$  auf den erhaltenen Output.
7: Setze  $\tilde{w}$  auf den Anteil von  $x_o$ , der die Lifting-Variablen repräsentiert
8: while  $(\|\tilde{w}\|_{\infty} \geq \hat{\varepsilon})$  und  $(x_o$  war zulässig) do
9:    $\pi := \pi \cdot 100$ 
10:  if  $\pi > \pi_{\max}$  then
11:    Abbruch der Schleife
12:  end if
13:  Rufe Alg. C.1 bzw. C.2 für das Problem mit den Parametern  $w, \pi$  sowie
    Startlösung  $(x_o)_0$  auf und setze  $x_o$  und  $fval$  auf den erhaltenen Output.
14:  Setze  $\tilde{w}$  auf den Anteil von  $x_o$ , der die Lifting-Variablen
15: end while

```

dingungen in der Startlösung $(x_u)_0$ zulässig sind. Dies geschieht im Algorithmus in den Zeilen 1 bis 5, indem w so lange verkleinert wird, bis eine zulässige Lösung des zugehörigen Problems gefunden wird. Das Finden einer solchen Lösung wird realisiert, indem zunächst aus l_{c_m} und u_{c_m} ein zulässiger Mautvektor c_m konstruiert wird. Dabei werden die Komponenten von c_m für die sowohl eine untere als auch eine obere Grenze existiert, auf das arithmetische Mittel aus beiden Werte gesetzt. Existiert nur eine untere beziehungsweise obere Grenze, so wird zu dieser Grenze 0,1 addiert respektive 0,1 subtrahiert. Alle anderen Komponenten werden auf 0 gesetzt. Mit diesem Mautvektor c_m wird anschließend mittels der MATLAB-Funktion *linprog* nach einer zulässigen Lösung für das mit dem Vektor w veränderte Problem der unteren Stufe (3.9) gesucht. Die Abfrage, ob eine zulässige Lösung gefunden wurde, erfolgt mittels den von *linprog* zurückgegebenen Exitflags.

Der zweite wichtige Aspekt bei der Lösung eines gelifteten Problems besteht darin, nach dem Finden einer Lösung des gelifteten Problems zu testen, ob das eingeführte w in allen Komponenten gleich 0 ist. Da die hier geforderte Genauigkeit in [9] nicht näher bestimmt ist, soll diese gleich der Genauigkeit der zu optimierenden Variable sein, die dem Löser, in diesem Fall *fmincon*, vorgegeben wird. Bei den Versuchen wurde diese auf 10^{-7} gesetzt. Ist das w nicht innerhalb dieser Genauigkeit gleich 0 und hat der Löser jedoch gleichzeitig eine zulässige Lösung gefunden, so muss der Parameter π vergrößert werden. Dies erfolgt in den Schritten 8-14 von Algorithmus 1. Überschreitet π den Parameter π_{\max} , der stets auf 10^{10} gesetzt wird, oder ist eine gefundene Lösung nicht zulässig, so

bricht die Schleife ab. Der Test, ob eine Lösung zulässig war, erfolgt wieder mittels der Exitflags, die die verwendete Lösungsroutine *fmincon* zurückgibt.

Der Startwert für den Parameter π wurde in [9] mit 1 angegeben. Um eine häufige Erhöhung dieses Parameters zu vermeiden und so die benötigte Rechenzeit zu verkleinern, wurde bei den Untersuchungen $\pi = 100$ gesetzt. Insgesamt erhielt man bei der Lösung der nach [9] modifizierten Probleme, dass in 69 von 72 Fällen bei denen durch die Algorithmen eine zulässige Lösung ausgegeben wurde, π mit dieser Wahl ausreichend groß war.

6.2. Ergebnisse der vorgestellten Beispiele

Bevor die Ergebnisse der nach Algorithmus B.1 erzeugten Probleme dargestellt werden, erfolgt nun zunächst die Präsentation der Resultate der vorgestellten Beispiele 2.1 und 2.2. Zur Berechnung wurde jeweils MATLAB 8.4.0.150421 (R2014b) eingesetzt. Als Genauigkeiten für die Zielfunktion, die Verletzung der Nebenbedingungen sowie als Schrittlänge der zu optimierenden Variable wurde jeweils 10^{-7} verwendet.

Darüber hinaus wurde zur Bestimmung des Grades der Verletzung der Nebenbedingungen einer gefundenen Lösung x_o^* , die Funktion

$$\max\text{Vio}(x_o^*) = \max\{\max\{0, g(x_o^*)\}, |h(x_o^*)|, |\min\{G(x_o^*), H(x_o^*)\}|\}$$

verwendet, die in [23] ebenfalls für diesen Zweck genutzt wurde. Ein Problem wird als gelöst betrachtet, wenn $\max\text{Vio}(x_o^*) \leq 10^{-6}$ gilt.

Für das Beispiel 2.1 erhält man die in Tabelle D.1 aufgeführten Resultate. Bei den Ergebnissen fällt auf, dass ohne Regularisierung stets ein optimaler Wert $(c_m)_7 = 0,1$ zurückgegeben wurde. Dieser ist jedoch Teil der Startlösung $(x_o)_0$, somit hat sich der Punkt x_o^* , welcher vom Algorithmus ausgegeben wurde, für c_m nicht von der Startlösung entfernt. Die anderen Kombinationen aus den zwei Formulierungen Problemformulierungen und den Regularisierungsverfahren gaben, falls eine zulässige Lösung erreicht wurde, ein $(c_m)_7$ aus der Menge $\{2, 3, 5\}$ sowie die dazugehörigen Zielfunktionswerte zurück. Wie aus der Darstellung der zugehörigen Zielfunktion in [18] deutlich wird, handelt es sich bei allen drei Lösungen sowie den zugehörigen Flüssen um lokale Minima des Problems. Bei der Anwendung des Regularisierungsverfahrens von Steffensen und Ulbricht wurde sowohl in der Formulierung mit als auch ohne Lift keine Lösung gefunden. Dies trifft auch auf das Verfahren von Lin und Fukushima in der Formulierung ohne Lift zu. Während die Genauigkeiten der Lösungen bezüglich der Verletzung der Nebenbedingungen alle ähnliche Resultate lieferten, benötigten die Regularisierungen zur Lösung der Formulierung mit Lift deutlich mehr Zeit, insbesondere die Regularisierung von Scholtes.

Die Ergebnisse der Berechnungen an Beispiel 2.2 sind in Tabelle D.2 dargestellt. Auch hier fällt auf, dass sich das Verfahren ohne Regularisierung in $(c_m)_2$ nicht von der Startlösung entfernt hat. Darüber hinaus lieferten alle Regularisierungsverfahren sowohl in der Formulierung mit als auch ohne Lift einen Wert von $(c_m)_2 > 1$ sowie einen Funktionswert

kleiner als die eingestellte Funktionswerttoleranz von 10^{-7} . Auch die zur Berechnung der Resultate benötigten Zeiten sind vergleichbar.

6.3. Resultate der in Kapitel 5 generierten Beispiele

Der Vergleich der verschiedenen numerischen Ansätze erfolgt nun anhand der 40 Beispiele, welche in Kapitel 5 thematisiert wurden. Die Berechnung erfolgte wieder mittels MATLAB 8.4.0.150421 (R2014b) und die für das Verfahren benötigten Genauigkeiten wurden, wie bereits dargestellt, wieder auf 10^{-7} gesetzt.

Zur Darstellung der Ergebnisse sollen die Performance-Profile verwendet werden, welche von Dolan und Moré in [14] eingeführt wurden und sowohl in [9] als auch in [23] ebenfalls zur Analyse der Ergebnisse genutzt wurden. Die Analyse soll dabei bezüglich der Verletzung des Nebenbedingungen, des Zielfunktionswertes in dem ausgegebenen Punkt sowie der zur Berechnung benötigten Zeit erfolgen. Die Berechnung der Verletzung der Nebenbedingungen erfolgte dabei wieder mit der bereits vorgestellten Funktion $\max\text{Vio}(x_o^*)$. Anders als in [23] ist jedoch die Modifizierung des ursprünglichen Ansatzes zur Auswertung der Zielfunktion nicht notwendig, da die Zielfunktionswerte hier stets nichtnegativ sind. Um realistische Ergebnisse zu erhalten, wird dabei für die Untersuchung des Zielfunktionswertes und der benötigten Zeit die zu untersuchende Größe auf U unendlich gesetzt, falls die ermittelte Lösung nicht in der geforderten Genauigkeit zulässig ist, also $x_o^* \max\text{Vio}(x_o^*) \leq 10^{-6}$ nicht erfüllt. Die Performance-Profile wurden gebildet, indem zunächst für jedes untersuchte Optimierungsproblem das Minimum der untersuchten Größe über die Ergebnisse aller Kombinationen aus den Regularisierungen und dem Ansatz mit bzw. ohne Lift gebildet wurde. Anschließend wurde für jede Kombination untersucht, für welchen Anteil der Probleme das durch diese Kombination ermittelte Ergebnis innerhalb des q -fachen des jeweiligen Minimalwertes liegt, wobei der Faktor q schrittweise erhöht wird. Daraus folgt, dass je schneller der Graph einer Kombination steigt, um so mehr Lösungen dieser Kombination bezüglich der betrachteten Größe nahe an dem jeweiligen Minimum liegen.

Betrachtet man den Vergleich der Zielfunktionswerte in Abbildung 3, so fällt auf, dass sowohl in der Formulierung mit Lift als auch ohne Lift das Lösungsverfahren ohne Regularisierung alle Probleme löst. Gleichzeitig steigen die zugehörigen Graphen nur sehr langsam an, was drauf hindeutet, dass die Zielfunktionswerte der gefundenen Lösungen größer als die anderer gefundener Lösungen sind. Betrachtet man die zur Berechnung der Lösung notwendigen Iterationen, so fällt auf, dass bei der Formulierung ohne Lift stets nur sehr wenige Iterationen durchgeführt wurden. So wurde bei der Formulierung ohne Lift in 25 Fällen nach nur einer Iteration ein Ergebnis zurückgegeben und gleichzeitig wurden bei nur 2 Problemen mehr als 5 Iterationen benötigt. Ein ähnliches Verhalten ist bei den Problemformulierungen mit Lift zu beobachten. Obwohl hier der Parameter w verkleinert werden muss, bis dieser der Forderung $\|w\|_\infty \leq 10^{-7}$ genügt, wurden auch hier nur in 5 Fällen mehr als 5 Iterationen benötigt.

Insgesamt benötigte dieses Verfahren also sehr wenige Iterationen, gab aber auch Lö-

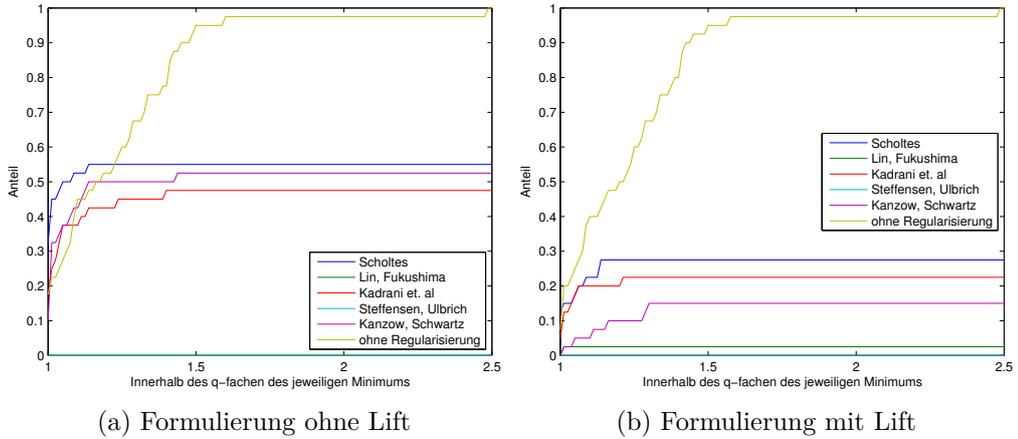


Abbildung 3.: Vergleich der berechneten Zielfunktionswerte

sungen mit verhältnismäßig schlechten Zielfunktionswerten aus. Abgesehen von der Lösung ohne Regularisierung, lieferte der Ansatz von Scholtes sowohl mit als auch ohne Lift die besten Resultate, dicht gefolgt von den Verfahren von Kanzow und Schwartz beziehungsweise von Kadrani et al. Dabei erhielt man mit der Regularisierung von Kanzow und Schwartz ohne Lift bessere Ergebnisse, wohingegen das Verfahren von Kadrani et al. mit Lift bessere Resultate ergab. Die Verfahren von Lin und Fukushima sowie von Steffensen und Ulbricht lieferten nur eine beziehungsweise keine Lösung.

Vergleicht man die Ergebnisse der Formulierungen mit Lift mit denen der Formulierung ohne Lift, so stellt man fest, dass die Verwendung des Liftes für diese Probleme die numerischen Ergebnisse insgesamt verschlechterten.

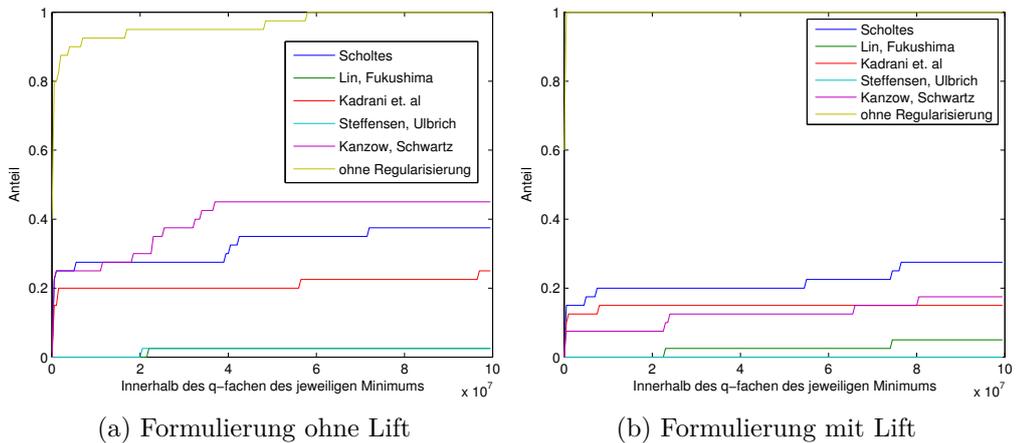


Abbildung 4.: Vergleich der Verletzung der Nebenbedingungen

Bei der Auswertung der Performance-Profile für die Verletzung der Nebenbedingungen

fällt auf, dass auch hier der Ansatz ohne Regularisierung die besten Werte lieferte. Darüber hinaus lieferten die Regularisierungen von Scholtes, Kanzow und Schwartz sowie von Kadrani et al. die besten Resultate. Mit dem Ansatz von Kanzow und Schwartz erhielt man die höchsten Genauigkeiten für Formulierung ohne Lift, wohingegen bei der Formulierung mit Lift die Regularisierung von Scholtes die besten Ergebnisse lieferte. Die schlechten Resultate der Regularisierungsverfahren von Steffensen und Ulbrich sowie von Lin und Fukushima zeigten sich auch an dieser Stelle. Insgesamt fällt auch hier wieder auf, dass die Ergebnisse der Formulierung ohne Lift besser als die der Formulierung mit Lift waren.

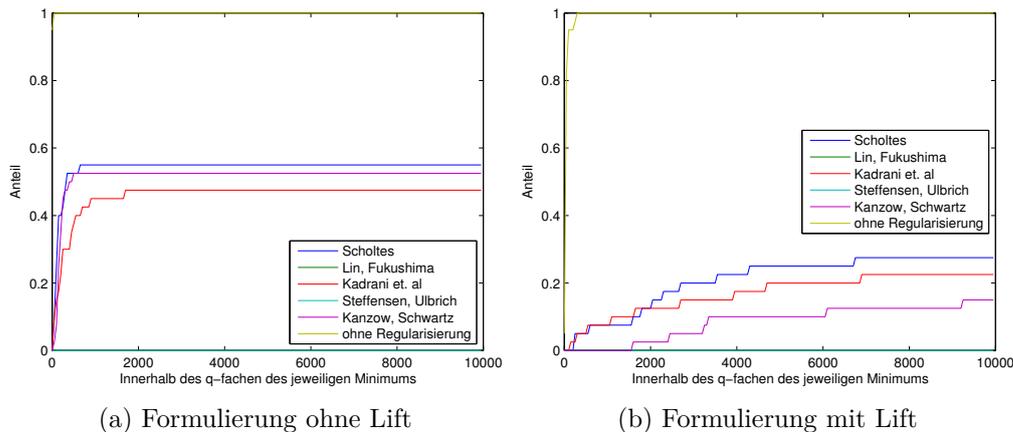


Abbildung 5.: Vergleich der benötigten Zeit

Die Darstellung der für die Berechnung der jeweiligen Lösung benötigten Zeit verdeutlicht, dass der Ansatz ohne Regularisierung deutlich schneller eine Lösung lieferten als die Verfahren mit Regularisierung. Dies war aufgrund der wenigen benötigten Iterationen in diesen Fällen auch zu erwarten. Wurde die Formulierung ohne Lift verwendet, so lagen die Regularisierungen von Scholtes und Kanzow/Schwartz nahezu gleichauf, wobei mit dem Verfahren von Scholtes noch leicht bessere Ergebnisse erzielt wurden. Das Anwenden des Lifts führte dazu, dass die besseren Ergebnisse von den Regularisierungen von Scholtes und Kadrani et al. erzielt wurden. Bei der Betrachtung der benötigten Zeit wird der Unterschied zwischen der Formulierung mit und der Lift ohne besonders deutlich, wobei zur Lösung eines gelifteten Problems deutlich mehr Zeit benötigt wurde.

Beim weiteren Analysieren der Ergebnisse stellt man fest, dass unabhängig davon ob der Lift verwendet wird oder nicht, der Anteil der Probleme, die mit den Verfahren gelöst wurden, generell stark abnahm. So wurden, mit dem Regularisierungsverfahren von Scholtes 15 von den 20 kleineren Problemen (vgl. Tabelle 3) ohne Lift gelöst, jedoch bei leicht größeren Problemen nur noch 8 der 20 Aufgaben. Wurden die Probleme mit dem Lifting-Verfahren von Hatz et al. modifiziert, so wurden nur noch 6 beziehungsweise 4 der Probleme gelöst. Im Weiteren soll eine Diskussion der gewählten Modellierung und den vorgestellten Ergebnissen erfolgen.

7. Diskussion der Ergebnisse

Zunächst soll nun eine Einschätzung der gewählten Modellierung des Problems erfolgen. Ausgehend von den im Modell getroffenen Annahmen an das Verhalten der Fahrzeugführer, stellt man fest, dass sich die mit diesem Modell ermittelten Ergebnisse nicht unbedingt auf praktisch auftretende Situationen übertragen lassen. Insbesondere die Annahme, nach der sich die Fahrzeugführer bei verschiedenen gesamtkostenminimalen Flüssen für denjenigen entscheiden, der für den Betreiber des Mautnetzwerkes die besten Resultate liefert, ist praktisch zweifelhaft, insbesondere da den Straßennutzern nicht zwangsläufig das Ziel des Betreibers bekannt sein muss. In einem solchen Fall kann die getroffene Annahme nicht erfüllt werden. Allerdings gibt es selbst wenn die gewünschte Verteilung der Restkapazitäten bekannt ist, praktisch keine Garantie, dass sich die Fahrzeugführer in diesem Sinne verhalten, da die Alternativen ebenfalls gesamtkostenminimal sind.

Weitere Probleme bei der Anwendung des Modells auf praktische Probleme ergeben sich, wenn die Maut nicht für alle Verkehrsteilnehmer identisch sein soll oder die Maut einer Kante von der genutzten Kapazität einer Kante abhängen soll. Tritt ein solcher Fall ein, muss das Modell erweitert werden.

Eine weitere sich ergebende Schwierigkeit ist, dass Flüsse einer Lösung des Problems der unteren Stufe und somit auch des Modells insgesamt nicht ganzzahlig sein müssen, wie beispielsweise in [16, Kapitel 17] dargestellt wird. Wird für eine Anwendung die Ganzzahligkeit der Lösung gefordert, müssen weitere Methoden der diskreten Optimierung angewendet werden.

Ein weiteres mögliches Problem der Modellierung ergibt sich, wenn untersucht wird, wie gut die Lösungen in Abhängigkeit von der Größe des Graphen und der Anzahl der Transporte sind. Dabei konnte festgestellt werden, dass die Anzahl der gelösten Probleme stark gesunken ist, wenn diese beiden Problemparameter erhöht wurden. Eine mögliche Erklärung für dieses Verhalten ist, dass bei der gewählten Modellierung die Anzahl der Nebenbedingungen bei Steigerung der jeweiligen Größen schnell steigt. Dieses Problem der gewählten, Kanten-basierten Modellierung wurde in [16, Abschnitt 17.5] erläutert. Als alternativen Ansatz wurde eine Pfad-basierte Modellierung vorgeschlagen, bei der für jede Ware vor der eigentlichen Berechnung die möglichen Wege von jeder Quelle zu jeder Senke bestimmt werden und in der Modellierung schließlich nur die Flüsse entlang dieser Wege betrachtet werden. Dies verringert die Anzahl der Nebenbedingungen, kann aber insbesondere dann zu einer exponentiell wachsenden Anzahl an Variablen führen, wenn es für jedes Gut mehrere Quellen und Senken gibt. In [16, Abschnitt 17.5] wird erläutert wie dieses Problem umgangen werden kann.

Im Weiteren sollen die in Kapitel 6 vorgestellten Ergebnisse mit denen aus [23] und

[9] verglichen werden. Generell ist zur Vergleichbarkeit der Ergebnisse aber anzumerken, dass unterschiedliche numerische Lösungsverfahren verwendet wurden. Während in [23] das SQP-Verfahren des Löfers *snopt* der Umgebung TOMLAB verwendet wurde und in [9] ein filterSQP-Verfahren zum Einsatz kam, erfolgten die numerischen Untersuchungen in dieser Arbeit mittels der SQP-Implementation von MATLAB, sodass generell Unterschiede auftreten können. Gleichzeitig wurden sowohl in [23] als auch in [9] die für die Lösung verwendeten Parameter, mit denen die Genauigkeiten der Zielfunktion sowie der Verletzung der Nebenbedingungen und der zu optimierenden Variable vorgegeben wurden, nicht näher dargestellt. Demzufolge können die hier verwendeten Parameter von denen aus [9] und [23] abweichen, sodass sich auch dadurch Unterschiede ergeben können.

Zunächst soll ein Vergleich der Ergebnisse aus [9] mit den hier erhaltenen erfolgen. In [9] wurde untersucht, wie sich ein filterSQP-Verfahren während der Lösung eines MPCCs mit beziehungsweise ohne Lift verhält, falls keine Regularisierung verwendet wird. Dabei wird das MPCC als nichtlineare Optimierungsaufgabe in der Formulierung (4.2) gelöst. Vergleichbare Berechnungen wurden in dieser Arbeit ebenfalls vorgenommen. In Kapitel 6 zeigten die Ergebnisse, dass dabei stets nach wenigen Iterationen eine Lösung ausgegeben wurde. Das detaillierte Performance-Profil der benötigten Iterationen ist nun in Abbildung 6 dargestellt.

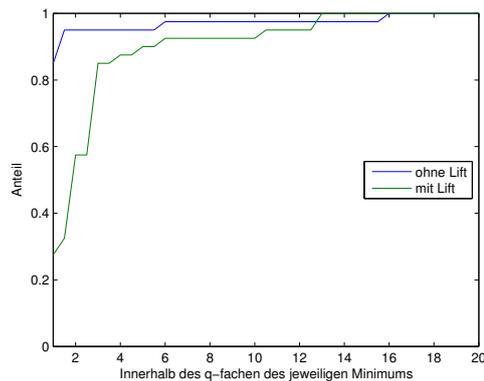


Abbildung 6.: Vergleich der Verletzung der Nebenbedingungen

In dieser Abbildung ist zunächst wieder die schon erwähnte Tatsache zu sehen, dass jeweils alle getesteten Probleme gelöst wurden. Weiterhin wird deutlich, dass das Lösen der Probleme ohne Lift weniger Iterationen benötigte. Dies war in Anbetracht der Ergebnisse aus Kapitel 6 auch zu erwarten, da bei dieser Formulierungsvariante sehr oft nur die Startlösung wieder zurückgegeben wurde. Vergleicht man diese Tatsache mit den Ergebnissen aus [9], so stellt sich heraus, dass bei den Problemen der Problemsammlung [17], die ebenfalls auf Transportproblemen beruhen und für die eine Lösung ermittelt werden konnte, also beispielsweise den Problemen *tollmpec* und *tap-09*, ebenfalls keine Verbesserung der Anzahl der Iterationen durch die Anwendung des Lifts erreicht werden konnte. Für die Probleme *tap-15* und *tollmpec1*, die ebenfalls Bilevel-Transportprobleme

sind, konnte in [9] keine Lösung ermittelt werden beziehungsweise nur eine für die Formulierung mit Lift, sodass keine Aussage darüber getroffen werden kann, ob die Anwendung des Lift die Anzahl der benötigten Iterationen reduziert wurde. Weiterhin muss an dieser Stelle noch einmal hervorgehoben werden, dass die ohne Regularisierung ermittelten Lösungen einen Zielfunktionswert hatten, der, außer bei einem Problem, stets größer als der von allen anderen eventuell ermittelten zulässigen Lösungen war. Aus diesem Grund liefern diese Verfahren zwar schnell gültige Lösungen, die aber bei der praktischen Suche nach einer Verteilung der Maut im Netzwerk kaum Vorteile bringen.

Nun soll ein Vergleich der erhaltenen Ergebnisse mit denen aus [23] erfolgen. Dabei ist zunächst anzumerken, dass die einzelnen Ergebnisse für die gestesten Probleme in [23] nicht angegeben sind. Somit kann an dieser Stelle nur ein Vergleich mit den Ergebnissen eines Querschnitts über verschiedene Problemtypen erfolgen, was dazu führt, dass die Vergleichbarkeit nur sehr begrenzt gegeben ist. Weiterhin ist zu beachten, dass in [23] die Probleme nur ohne die Anwendung eines Lifts gelöst wurden. Da es sich jedoch bei der Formulierung mit Lift nach wie vor um ein MPCC handelt, soll der Vergleich zu den Ergebnissen aus [23] im Weiteren zu den hier ermittelten Resultaten sowohl der Formulierung mit als auch ohne Lift erfolgen. Für die gleichzeitige Anwendung von Lift und Regularisierungsverfahren ist bisher keine Literatur bekannt. In [23] wurde deutlich, dass bezüglich der erreichten Zielfunktionswerte und der benötigten Zeit die Regularisierungen von Kadrani et. al, Steffensen und Ulbrich sowie von Kanzow und Schwartz vergleichbare Ergebnisse lieferten. Die Regularisierung von Scholtes lieferte wiederum leicht bessere Resultate, während der Ansatz von Lin und Fukushima eine deutlich schlechtere Performance zeigte. Beim Vergleich der erzeugten Genauigkeiten stellt man fest, dass die besten Resultate von den Regularisierungen von Steffensen und Ulbrich sowie von Scholtes erbracht wurden. Danach folgten die Ansätze von Kanzow und Schwarz beziehungsweise von Kadrani et al. Die schlechtesten Ergebnisse wurden auch hier von dem von Lin und Fukushima vorgestellten Verfahren erzielt. Vergleicht man diese Ergebnisse mit den in dieser Arbeit ermittelten Resultaten, so stellt sich heraus, dass der Ansatz von Scholtes, der in [23] mit die besten Resultate lieferte, an dieser Stelle sowohl mit als auch ohne Lift die weiteren verwendeten Verfahren übertrifft. Einzig bei der Genauigkeit der Formulierung ohne Lift wurde das Verfahren von dem Ansatz nach Kanzow und Schwartz übertroffen. Insgesamt kann man weiterhin sagen, dass eben jenes Verfahren von Kanzow und Schwartz in der Formulierung ohne Lift etwas besser funktionierte als das Verfahren von Kadrani et al. Betrachtet man die Formulierung mit Lift, so ist es genau anders herum. Somit decken sich die Ergebnisse mit Beobachtungen aus [23]. Betrachtet man die Regularisierung von Lin und Fukushima, so konnte mit dieser bei den in Kapitel 6 vorgestellten Untersuchungen kein Problem gelöst werden, wenn die Problemformulierung ohne Lift Anwendung fand. Wurde der Lift verwendet, so erreichte der Lösungsalgorithmus nur eine Lösung, die die geforderte Genauigkeit der Verletzung der Nebenbedingungen aufwies. In [23] lieferte die Regularisierung nach Lin und Fukushima die schlechtesten Ergebnisse, was sich folglich mit den Resultaten dieser Arbeit deckt. Mögliche Erklärungsansätze für die schlechten numerischen Eigenschaften werden in [23] geliefert, indem dargestellt wird, dass die beiden durch das Verfahren

eingeführten Nebenbedingungen für sehr kleine Regularisierungsparameter t numerisch fast zusammenfallen und auf diese Weise Schwierigkeiten verursacht haben könnten. Die größten Unterschiede zu den Ergebnissen aus [23] ergeben sich beim Verfahren von Steffensen und Ulbrich. Während dieser Ansatz in [23] bei der Genauigkeit der Lösung mit Abstand die besten Resultate lieferte, konnte in dieser Arbeit damit kein einziges Problem gelöst werden. Weitere Untersuchungen ergaben die im Anhang E dargestellten Ergebnisse. Dabei wurden die in Kapitel 6 verwendeten Probleme leicht modifiziert, indem auf jeder Kante eine Mindestmaut gelten soll. Berechnet man die Lösungen dieser Beispiele, so werden mit dem Regularisierungsverfahren von Steffensen und Ulbrich in der Formulierung ohne Lift 32,5 % der Probleme und mit Lift 40 % der Probleme gelöst werden. Während im ersten Fall das Verfahren von Steffensen und Ulbrich dennoch neben dem Ansatz von Lin und Fukushima die schlechtesten Resultate lieferte, so ergaben sich im Fall mit Lift die besten Resultate aller verwendeten Regularisierungen. Dies zeigt sich besonders an der auch schon in [23] beobachteten hohen Genauigkeiten der Lösungen. Gleichwohl bedeuten diese modifizierten Beispiele aber eine Einschränkung der möglichen modellierbaren Probleme, sodass die Ergebnisse dieser Beispiele nicht weiter im Detail betrachtet werden sollen.

Betrachtet man die Ergebnisse der Formulierungen mit und ohne Lift im Vergleich, so stellt man insgesamt fest, dass bei den verwendeten Beispielen, sowohl bei der Verwendung einer Regularisierung als auch ohne eine solche, keine Verbesserung der Resultate eintritt. Eine Erklärung dafür kann sein, dass die ursprüngliche Intention der Einführung des Lift, die die Sicherstellung der MPCC-LICQ beabsichtigte, zumindest bei der in dieser Arbeit gewählten Formulierung nicht in jedem Fall anwendbar ist, wie in Kapitel 3 gezeigt wurde. In den meisten Fällen war die MPCC-LICQ in den erhaltenen Lösungen aber dennoch erfüllt. Gleichzeitig führt die Einführung der Modifikation der Probleme nach [9] aber in jedem Fall zu einer Erhöhung der Problemdimension und somit dazu, dass ein Problem potenziell schwieriger zu lösen ist.

Ein weiterer wichtiger Punkt bei der Frage, inwieweit diese Ergebnisse auf beliebige praktisch auftretende Probleme übertragen werden können, zielt letztlich darauf ab, wie allgemeingültig die getesteten Probleme sind. Dazu ist zunächst generell anzumerken, dass die Verfahren nur an sehr kleinen Problemen getestet wurden. Dabei hat sich jedoch gezeigt, dass die tendenziellen Unterschiede zwischen den Verfahren hinsichtlich ihrer Resultate erhalten blieben, wenn die zugrundeliegenden Graphen und damit auch die Optimierungsaufgaben vergrößert wurden. Weiterhin wurden bei den Problemen stets auf allen Kanten eine Maut zugelassen und diese auch nicht nach oben beschränkt. An dieser Stelle können in praktischen Situationen andere Bedingungen an die Maut auftreten, was die Resultate verändern kann, wie an den zusätzlichen Untersuchungen im Anhang E deutlich wurde. Auch die Anzahl von Gütern kann bei praktischen Anwendungen deutlich von der abweichen, die in den hier verwendeten Beispielen angenommen wurde.

8. Conclusion

In dieser Arbeit wurden MPCCs untersucht, die sich ergeben, wenn man die Annäherung eines Flusses auf einem Netzwerk an eine vorgegebene Verteilung mittels der Einführung einer Maut erreichen möchte.

Dabei wurde die in [9] vorgeschlagene Modifikation eines MPCCs angewendet, die sicherstellen soll, dass unter gewissen Annahmen an das Problem der unteren Stufe, die MPCC-LICQ für das Problem der oberen Stufe erfüllt ist. Es konnte gezeigt werden, dass dieses Resultat nicht auf die hier betrachteten MPCCs angewendet werden kann. Darüber hinaus wurde diese Modifikation sowie die in [23] vorgeschlagenen Regularisierungsverfahren in MATLAB implementiert und anhand von erstellten Beispielproblemen getestet. Dabei ergab sich, dass die Regularisierung nach Scholtes die besten Resultate lieferte. Generell wurden bessere Resultate erzielt, wenn keine Modifikation der Probleme nach [9] vorgenommen wurde.

Literatur

- [1] A. Kadrani, J. Dussault and A. Benchakroun. „A New Regularization Scheme for Mathematical Programs with Complementarity Constraints“. In: *SIAM Journal on Optimization* 20.1 (2009), S. 78–103. DOI: [10.1137/070705490](https://doi.org/10.1137/070705490).
- [2] L. T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes (MPS-SIAM Series on Optimization)*. Philadelphia: SIAM-Society for Industrial und Applied Mathematics, 2010.
- [3] Bundesministerium der Justiz und für Verbraucherschutz. *Bundesfernstraßenmautgesetz vom 12. Juli 2011 geändert durch Artikel 2 Absatz 152 des Gesetzes vom 7. August 2013*. 07.08.2013. URL: <http://www.gesetze-im-internet.de/bundesrecht/bfstrmg/gesamt.pdf>.
- [4] CDU Deutschlands, CSU-Landesleitung und SPD. *Deutschlands Zukunft gestalten. Koalitionsvertrag zwischen CDU, CSU und SPD*. 2013. URL: <https://www.cdu.de/sites/default/files/media/dokumente/koalitionsvertrag.pdf>.
- [5] N. Deo. *Graph theory with applications to engineering and computer science*. Prentice-Hall Series in Automatic Computation. Englewood Cliffs: Prentice-Hall Inc., 1974.
- [6] Y. Chen and M. Florian. „The nonlinear bilevel programming problem: formulations, regularity and optimality conditions“. In: *Optimization. A Journal of Mathematical Programming and Operations Research* 32.3 (1995), S. 193–209. DOI: [10.1080/02331939508844048](https://doi.org/10.1080/02331939508844048).
- [7] G.-H. Lin and M. Fukushima. „A Modified Relaxation Scheme for Mathematical Programs with Complementarity Constraints“. In: *Annals of Operations Research* 133.1-4 (2005), S. 63–84. DOI: [10.1007/s10479-004-5024-z](https://doi.org/10.1007/s10479-004-5024-z).
- [8] K. Schmidt und G. Trenkler. *Einführung in die moderne Matrix-Algebra. Mit Anwendungen in der Statistik*. 2. Aufl. Berlin: Springer, 2006. DOI: [10.1007/3-540-33008-9](https://doi.org/10.1007/3-540-33008-9).
- [9] K. Hatz, S. Leyffer, J. P. Schlöder and H. G. Bock. „Regularizing Bilevel Nonlinear Programs by Lifting“. Preprint ANL/MCS-P4076-0613. 2013.
- [10] D. A. Harville. *Matrix algebra from a statistician's perspective*. New York: Springer-Verlag, 1997.
- [11] J. Matoušek und J. Nešetřil. *Diskrete Mathematik - Eine Entdeckungsreise*. 2. Aufl. Berlin, Heidelberg, New York: Springer-Verlag, 2007.
- [12] B. Korte und J. Vygen. *Kombinatorische Optimierung*. 2. Aufl. Heidelberg: Springer Spektrum, 2012.

- [13] A. Beutelsbacher und M.-A. Zschiegner. *Graphentheorie*. Wiesbaden: Vieweg + Teubner, 2007. DOI: [10.1007/978-3-8348-9182-2_8](https://doi.org/10.1007/978-3-8348-9182-2_8).
- [14] E.D. Dolan and J.J. Moré. „Benchmarking optimization software with performance profiles“. In: *Mathematical Programming* 91.2 (2002), S. 201–213. DOI: [10.1007/s101070100263](https://doi.org/10.1007/s101070100263).
- [15] O. Sluciak. *Graph adjacency matrix to incidence matrix*. 2011. URL: www.mathworks.com/matlabcentral/fileexchange/24661-graph-adjacency-matrix-to-incidence-matrix.
- [16] R. K. Ahuja, T. L. Magnanti and James B. Orlin. *Network flows*. Theory, algorithms, and applications. Englewood Cliffs: Prentice Hall Inc., 1993.
- [17] S. Leyffer. *MacMPEC. AMPL collection of MPECs*. 2012. URL: <http://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC>.
- [18] S. Dempe und S. Lohse. „Optimale Mautgebühren“. In: Bernd Luderer (Hrsg.). *Die Kunst des Modellierens*. Hrsg. von Bernd Luderer. Mathematisch-ökonomische Modelle. Wiesbaden: Vieweg+Teubner, 2008.
- [19] M. Ulbrich und S. Ulbrich. *Nichtlineare Optimierung*. Basel: Birkhäuser, 2012.
- [20] R. Fletcher, S. Leyffer, D. Ralph and Stefan Scholtes. „Local convergence of SQP methods for mathematical programs with equilibrium constraints“. In: *SIAM Journal on Optimization* 17.1 (2006), S. 259–286. DOI: [10.1137/s10526-234-0240-7382](https://doi.org/10.1137/s10526-234-0240-7382).
- [21] S. Scholtes. „Convergence Properties of a Regularization Scheme for Mathematical Programs with Complementarity Constraints“. In: *SIAM Journal on Optimization* 11.4 (2001), S. 918–936. DOI: [10.1137/S1052623499361233](https://doi.org/10.1137/S1052623499361233).
- [22] C. Kanzow and A. Schwartz. „A New Regularization Method for Mathematical Programs with Complementarity Constraints with Strong Convergence Properties“. In: *SIAM Journal on Optimization* 23.2 (2013), S. 770–798. DOI: [10.1137/100802487](https://doi.org/10.1137/100802487).
- [23] T. Hoheisel, C. Kanzow and A. Schwartz. „Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints“. In: *Mathematical Programming* 137.1-2 (2013), S. 257–288. DOI: [10.1007/s10107-011-0488-5](https://doi.org/10.1007/s10107-011-0488-5).
- [24] R. Reinhardt, A. Hoffmann und T. Gerlach. *Nichtlineare Optimierung*. Berlin, Heidelberg: Springer, 2013. DOI: [10.1007/978-3-8274-2949-0_4](https://doi.org/10.1007/978-3-8274-2949-0_4).
- [25] S. Steffensen and M. Ulbrich. „A New Relaxation Scheme for Mathematical Programs with Equilibrium Constraints“. In: *SIAM Journal on Optimization* 20.5 (2010), S. 2504–2539. DOI: [10.1137/090748883](https://doi.org/10.1137/090748883).
- [26] J. Nocedal and S.J. Wright. *Numerical optimization*. 2. edition. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006.

- [27] Eberhard Zeidler. „Analysis“. In: Eberhard Zeidler (Hrsg.). *Springer-Taschenbuch der Mathematik*. Hrsg. von Eberhard Zeidler. 3. Aufl. Heidelberg: Springer Spektrum, 2013. DOI: [10.1007/978-3-8348-2359-5](https://doi.org/10.1007/978-3-8348-2359-5).
- [28] Eberhard Zeidler. „Vektoralgebra“. In: Eberhard Zeidler (Hrsg.). *Springer-Taschenbuch der Mathematik*. Hrsg. von Eberhard Zeidler. 3. Aufl. Heidelberg: Springer Spektrum, 2013. DOI: [10.1007/978-3-8348-2359-5](https://doi.org/10.1007/978-3-8348-2359-5).

Anhang A.

Herleitung der Lösungen von Beispiel 2.2

Im Folgenden sollen alle Lösungen von Beispiel 2.2 ermittelt werden.

Sei dazu $c_m^* = (0 \ (c_m)_2 \ 0)^\top$ mit $(c_m)_2 \geq 1$. Für den Transport von Gut 1 und 2 sind die in Tabelle A.1 dargestellten Pfade möglich.

Pfad	Kosten	maximale Kapazität
	c_{W_i}	u_{W_i}
$W_1 = (1, 2, 3)$	2	6
$W_2 = (1, 3)$	$1 + (c_m)_2$	6

Tabelle A.1.: Mögliche Pfade

Nun sollen alle gesamtkostenminimalen Lösungen bestimmt werden. Dazu wird ein Parameter $\lambda \in [0, 1]$ eingeführt, mit dem die verschiedenen Lösungen beschrieben werden sollen.

Sei nun zunächst $(c_m)_2 > 1$. Die sich daraus ergebenden gesamtkostenminimalen Flüsse können Tabelle A.2 entnommen werden. In dieser ist für jeden Pfad dargestellt, welche Ware diesen Pfad mit welcher Kapazität nutzt.

Pfad	Kosten	max. Kapazität	Ware	genutzte Kapazität
W_1	2	6	1	$4 - 2\lambda$
			2	$2 + 2\lambda$
W_2	> 2	6	1	2λ
			2	$2 - 2\lambda$

Tabelle A.2.: Optimaler Fluss

Gilt $(c_m)_2 = 1$, so gilt $c_{W_1} = c_{W_2}$, woraus sich neben der in Tabelle A.2 dargestellten Lösungen noch weitere gesamtkostenminimale Flüsse ergeben. Durch die gewählte Modellierung verhalten sich die Gütertransporte jedoch stets so, dass bei identischen Gesamtkosten für den Fluss stets derjenige verwendet wird, der die Restkapazitäten an x_z am weitesten annähert, also die Zielfunktion des Problems der oberen Stufe minimal wird. Im Weiteren wird sich aber zeigen, dass die Zielfunktion genau bei den in

Tabelle A.2 beschriebenen Flüssen minimal wird. Somit spielen die gesamtkostenminimalen Flüsse, die im Fall $(c_m)_2 = 1$ zusätzlich auftreten, im Problem der oberen Stufe keine Rolle und werden deshalb nicht weiter betrachtet. Wandelt man die beschriebenen Flüsse in die im Modell verwendete Schreibweise um, so erhält man

$$x^* = \begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} \quad \text{wobei} \quad x_1^* = \begin{pmatrix} 2 & 2 & 2 \end{pmatrix}^\top \quad \text{und} \quad x_2^* = \begin{pmatrix} 4 & 0 & 4 \end{pmatrix}^\top$$

und

$$x^{**} = \begin{pmatrix} x_1^{**} \\ x_2^{**} \end{pmatrix} \quad \text{wobei} \quad x_1^{**} = \begin{pmatrix} 4 & 0 & 4 \end{pmatrix}^\top \quad \text{und} \quad x_2^{**} = \begin{pmatrix} 2 & 2 & 2 \end{pmatrix}^\top$$

sowie die Konvexkombination $x_\lambda = \lambda x^* + (1 - \lambda)x^{**}$ für $\lambda \in [0, 1]$. Wertet man mit diesen Flüssen x_λ die Zielfunktion des Problems der oberen Stufe (2.8) aus, so erhält man

$$\begin{aligned} \|u - Jx_\lambda - x_r\|^2 &= \|u - J(\lambda x^* + (1 - \lambda)x^{**}) - x_r\|^2 \\ &= \|u - \lambda Jx^* - (1 - \lambda)Jx^{**} - x_r\|^2 \\ &= \|u - \lambda(x_1^* + x_2^*) - (1 - \lambda)(x_1^{**} + x_2^{**}) - x_r\|^2 \\ &\stackrel{x_1^* + x_2^* = x_1^{**} + x_2^{**}}{\implies} \|u - (x_1^* + x_2^*) - x_r\|^2 \\ &= \|x_r - x_r\|^2 \\ &= 0. \end{aligned}$$

Da die Zielfunktion als Norm stets nur nichtnegative Werte liefert, muss mit der beschriebenen Wahl von c_m und den draus folgenden optimalen Flüssen ein Minimum vorliegen. Betrachtet man die Formulierung (2.12) und setzt $w = 0$, was in einem Optimum stets gelten muss, so erhält man, dass mit den obigen Setzungen ebenfalls ein Minimum auftritt.

Die Tatsache, dass bereits alle Lösungen gefunden wurden, sieht man dran, dass wenn $0 \leq (c_m)_2 < 1$ gilt, daraus $c_{W_2} < c_{W_1}$ folgt. Somit würden sich in Tabelle A.2 die verwendeten Kapazitäten des Pfads W_1 mit dem W_2 vertauschen. Dies führt aber zu einem echt positiven Zielfunktionswert im Problem der oberen Stufe, was offensichtlich größer als der Zielfunktionswert im Fall $(c_m)_2 \geq 1$ ist und somit kein Optimum liefern kann.

Anhang B.

Algorithmus zur Erstellung der Beispielprobleme

Die Erstellung der zufälligen Probleme findet mit Hilfe des Algorithmus B.1 statt.

Algorithmus B.1 Erstellung der Beispielprobleme

Eingabe: ursprüngliche Knotenanzahl $\hat{n} \in \mathbb{N}^+$, Anzahl der Transporte $k \in \mathbb{N}^+$, Parameter $p \in (0, 2)$

Ausgabe: \mathcal{N} , c , b^l für $l \in K$, x_z , l_{c_m} , u_{c_m}

- 1: $N_1 := \{1, \dots, n\}$
 - 2: Erzeuge zufällige Punkte P_i für $i \in N_1$ mit $P_i \sim \mathcal{U}([0, 20] \times [0, 20])$.
 - 3: Entferne doppelte Punkte und verkleinere N_1 entsprechend
 - 4: $d_{i,j} := \frac{\text{dist}(P_i, P_j)}{\max_{e,f \in N_1} (\text{dist}(P_e, P_f))}$ für $j > i$ und $d_{i,j} = 0$ für $j \leq i$
 - 5: $q_{i,j} := 1 - d_{i,j} + z_{i,j}$ ($j > i$) mit $z_{i,j} \sim \mathcal{U}([0, 1])$
 - 6: Erstelle Adjazenzmatrix \mathcal{A} mit $a_{i,j} := \begin{cases} 1 & \text{falls } q_{i,j} > p \\ 0 & \text{sonst} \end{cases}$ für $j > i$
und $a_{i,j} = a_{j,i}$ für $j < i$ sowie $a_{i,j} = 0$ falls $i = j$
 - 7: **if** Graph mit Adjazenzmatrix \mathcal{A} ist nicht zusammenhängend **then**
 - 8: **return**
 - 9: **end if**
 - 10: **while** Falls in Zeile k von \mathcal{A} weniger als 2 Einträge **do**
 - 11: Lösche Zeile k sowie Spalte k aus \mathcal{A} , verkleinere N_1 entsprechend
 - 12: **end while**
 - 13: **if** $|N_1| < 0.7 \hat{n}$ **then**
 - 14: **return**
 - 15: **end if**
 - 16: $\mathcal{N} := \text{ADJ2INC}(\mathcal{A})$ ▷ Inzidenzmatrix
 - 17: $c_{i,j} := \frac{1}{10} \text{dist}(P_i, P_j)$ ▷ mautunabhängige Kosten
 - 18: $S = \emptyset$, $B = []$
 - 19: $m :=$ Anzahl der Spalten von \mathcal{N}
-

```

20: for  $l = 1, \dots, k$  do ▷ Knotenbilanzen
21:   Finde  $(i, j)$  sodass  $\text{dist}(P_i, P_j) = \max_{\substack{e, f \in N_1 \\ (e, f) \notin S}} (\text{dist}(P_e, P_f))$ 
22:    $S := S \cup \{(i, j)\}$ 
23:   Setze  $d^l \in \{1, \dots, 5\}$  zufällig und gleichverteilt
24:   Setze  $b^l \in \mathbb{R}^m$  sodass  $(b^l)_t := \begin{cases} -d, & \text{falls } t = i \\ d, & \text{falls } t = j \\ 0, & \text{sonst} \end{cases}$ 
25: end for
26:  $d_{\max} := \max_{l \in K} (d^l)$ 
27: Wähle  $u_{i,j}$  zufällig, gleichverteilt aus  $\{d_{\max}, \dots, 2d_{\max}\}$  ▷ Kantenkapazitäten
28: if bisheriges Problem nicht lösbar then
29:   return
30: end if
31:  $(x_z)_{i,j} = u_{i,j} \cdot z_{i,j}^2$  mit  $z_{i,j} \sim \mathcal{U}([0, 1])$  ▷ Restkapazitäten
32:  $l_{c_m} = \begin{pmatrix} 0 & \dots & 0 \end{pmatrix}^\top \in \mathbb{R}^m$  ▷ untere Beschränkung an Maut
33:  $u_{c_m} = \begin{pmatrix} \infty & \dots & \infty \end{pmatrix}^\top \in \mathbb{R}^m$  ▷ obere Beschränkung an Maut

```

Zu diesem Algorithmus folgen nun noch einige Anmerkungen.

- Bemerkung B.1.** • Die Berechnung der $q_{i,j}$ in Schritt 5 erfolgt derart, dass der auf $[0, 1]$ normierte Abstand $d_{i,j}$ von 1 abgezogen wird. Somit ist das zugehörige $q_{i,j}$ umso größer, je kleiner der Abstand der Punkte P_i und P_j ist. Nach dem Hinzufügen einer zufälligen Komponente erfolgt anschließend der Test, ob $q_{i,j}$ größer als ein vorher festgelegter Parameter p ist. Liegt dieser Fall vor, so wird die Kante zwischen den Punkten P_i und P_j sowie zwischen P_j nach P_i gesetzt. Somit werden insgesamt eher kürzere Kanten gesetzt. Beim Setzen der Quellen und Senken werden diese im Schritt 21 hingegen so ausgewählt, dass der Abstand zwischen diesen maximal wird. Dadurch wird insgesamt erreicht, dass die Transporte über mehrere Kanten stattfinden müssen und keine unmittelbare Verbindung zwischen den Quellen und Senken besteht.
- In den Zeilen 7 bis 9 wird getestet, ob der erzeugte Graph zusammenhängend ist. Der Test erfolgt dabei mittels der in [11, Hilfssatz 4.2.4] vorgestellten Tatsache, dass der i, j -te Eintrag der r -ten Potenz der Adjazenzmatrix angibt, wie viele Kantenzüge der Länge r zwischen den Knoten i und j existieren. Addiert man also die ersten $|N_1| - 1$ Potenzen von \mathcal{A} auf und erhält eine Matrix mit Einträgen, die gleich 0 sind, so ist der Graph nicht zusammenhängend. Dieses Resultat erhält man, da es in einem solchen Fall zwei Knoten gibt, die nicht durch einen Kantenzug mit maximaler Länge $|N_1| - 1$ verbunden sind. Da aber jeder Pfad ein Kantenzug ist und in einem Graph mit $|N_1|$ Knoten der längste Pfad maximal die Länge

$|N_1| - 1$ hat, kann der Graph in diesem Fall nicht zusammenhängend sein. Bei einer Implementierung kann die Addition der Potenzen von \mathcal{A} abgebrochen werden, wenn die Summe keine Null-Einträge mehr enthält.

- Die Schleife in den Zeilen 10 bis 12 stellt sicher, dass die Knoten ausgeschlossen werden, von denen nur eine Kante ausgeht bzw. an denen nur eine Kante ankommt. Solche Knoten haben auf den Transport in einem Netzwerk nur dann Einfluss, falls ein Transport bei diesen beginnt bzw. endet. In beiden Fällen liegt aber kein Netzwerkcharakter vor, da der Transport über diese eine eingehende bzw. ausgehende Kante stattfinden muss. Gleichzeitig wird aber durch die höhere Anzahl an Kanten die Problemdimension vergrößert. Aus diesem Grund werden derartige Knoten nicht weiter betrachtet.
- Die in Zeile 13 dargestellte Abfrage, ob nach dem Entfernen der Knoten, von denen weniger als 2 Kanten ausgehen, noch mindestens 70% der Knoten von der ursprünglich eingegebenen Knotenzahl \hat{n} erhalten geblieben sind, stellt sicher, dass der entstehende Graph nicht zu klein wird. Tritt dies auf, so wird kein Problem zurückgegeben.
- Die in Zeile 16 verwendete Funktion *ADJ2INC* stammt aus [15].
- In den Zeilen 28 bis 30 wird getestet, ob das bis dahin erstellte Problem lösbar ist. Dies erfolgt, indem zunächst aus den schon erzeugten Daten das Problem der unteren Stufe (2.3) generiert wird. Die von der Maut abhängigen Kantenkosten c_m werden dabei auf 0 gesetzt. Das entstandene lineare Optimierungsproblem wird mittels der MATLAB-Funktion *linprog* gelöst. Gibt die Routine dabei einen Exitflag aus, der anzeigt, dass keine Lösung gefunden wurde, so wird kein Problem zurückgegeben.

Anhang C.

Algorithmen zur Lösung regularisierter Probleme

An dieser Stelle sollen die in [23] beschriebenen Algorithmen zur Lösung von regularisierten Problemen sowie die verwendeten Parameter dargestellt werden. Generell wurde analog zu [23] $t_{\min} = 10^{-15}$ gesetzt.

Regularisation	S	LF	KDB	SU	KS
t_0	1	1	1	$\frac{2\pi}{\pi-2}$	1
σ	0,0001	0,01	0,01	0,01	0,01

Tabelle C.1.: Übersicht der verwendeten Parameter

Algorithmus C.1 Allgemeiner Algorithmus zum Lösen der reg. Probleme

Eingabe: Startlösung $(x_o)_0$, t_0 , σ , t_{\min}

Ausgabe: finale Iteration $x_{opt} = (x_o)_k$, zugehöriger Zielfunktionswert $fval$

- 1: $k := 0$
 - 2: **while** $t_k \geq t_{\min}$ und $(x_o)_k$ zulässig **do**
 - 3: Setze $(x_o)_{k+1}$, $fval$ auf die Ausgabe eines Minimierungsverfahrens für das mit t regularisierte Problem und $(x_o)_k$ als Startlösung.
 - 4: $k := k + 1$
 - 5: $t_k := \sigma \cdot t_k$
 - 6: **end while**
-

Algorithmus C.2 Verbesserter Algorithmus zum Lösen der reg. Probleme

Eingabe: Startlösung $(x_o)_0$, t_0 , σ , t_{\min}

Ausgabe: finale Iteration $x_{opt} = (x_o)_k$, zugehöriger Zielfunktionswert $fval$

- 1: $k := 0$
 - 2: **while** $t_k \geq t_{\min}$ und $(x_o)_k$ zulässig **do**
 - 3: Setze $(x_o)_{k+1}$, $fval$ auf die Ausgabe eines Minimierungsverfahrens für
 das mit t regularisierte Problem und $(x_o)_k$ als Startlösung.
 - 4: $k := k + 1$
 - 5: **if** $(x_o)_k$ unzulässig **then**
 - 6: **return**
 - 7: **end if**
 - 8: Setze $t_k := \min_{l \in \mathbb{N}} \{ \sigma^l \cdot t_{k-1} \mid x_o \text{ ist zulässig für das mit } \delta^l \cdot t_{k-1} \text{ regularisierte} \\ \text{Problem} \}$
 - 9: **end while**
-

Anhang D.

Ergebnisse der vorgestellten Beispiele

An dieser Stelle sollen die Resultate der numerischen Berechnungen der Beispiele 2.1 und 2.2 detailliert angegeben werden. Dabei soll $vio_{x_o} := \log(\max \text{Vio}(x_o^*))$ und $vio_w := \log(\|w^*\|_\infty)$ gelten.

Lift	Regularisierung	vio_{x_o}	vio_w	$(c_m)_7$	Zielfunktionswert	Zeit in s	p
nein	S	-3,88		3,00	47,75	0,14	
	LF	-0,61		2,74	14,00	0,30	
	KDB	-7,44		2,00	36,50	0,12	
	SU	-0,17		2,33	14,00	0,21	
	KS	-8,00		3,00	47,75	0,14	
	keine	-8,11		0,10	50,00	0,02	
ja	S	-7,52	-23,81	5,00	53,33	26,59	100
	LF	-7,30	-7,00	2,00	36,50	1,17	100
	KDB	-7,00	-7,00	2,00	36,50	3,73	100
	SU	-0,41	-14,21	2,61	14,00	7,30	100
	KS	-6,70	-7,00	3,00	47,75	2,40	100
	keine	-7,21	-20,42	0,10	50,00	0,04	100

Tabelle D.1.: Ergebnisse von Beispiel 2.1

Lift	Regularisierung	vio_{x_o}	vio_w	$(c_m)_2$	Zielfunktionswert	Zeit in s	p
nein	S	-8,00		1,01	7,89e-31	0,05	
	LF	-8,00		1,22	4,10e-22	0,07	
	KDB	-8,00		2,08	4,23e-17	0,06	
	SU	-15,05		1,00	7,89e-31	0,03	
	KS	-8,00		1,00	2,60e-16	0,05	
	ohne	-8,83		0,10	48,00	0,01	
ja	S	-8,00	-8,40	1,00	1,10e-17	0,04	100
	LF	-8,00	-8,58	1,24	4,28e-18	0,07	100
	KDB	-7,92	-8,72	1,72	6,32e-17	0,07	100
	SU	-15,82	-15,79	1,00	1,97e-31	0,03	100
	KS	-8,00	-8,17	1,00	9,51e-16	0,06	100
	ohne	-14,78	-22,41	0,01	48,00	0,02	100

Tabelle D.2.: Ergebnisse von Beispiel 2.2

Anhang E.

Ergebnisse bei modifizierten Beispiele

In diesem Anhang sollen kurz die Ergebnisse der modifizierten Probleme dargestellt werden. Der Anlass für diese weiteren Untersuchungen ist, dass das Regularisierungsverfahren nach Steffensen und Ulbrich in den Untersuchungen von [23] die besten Resultate erzielte und gleichzeitig von den in dieser Arbeit untersuchten Problemen kein einziges lösen konnte.

Im Weiteren werden wieder die Probleme aus Kapitel 6 zur Berechnung verwendet, allerdings wird stets $(l_{cm})_i = 1$ für alle $i \in A$ gesetzt. Löst man diese Probleme mit den selben Regularisierungen und Parametern, so erhält man die in den Abbildungen E.1 bis E.3 dargestellten Ergebnisse.

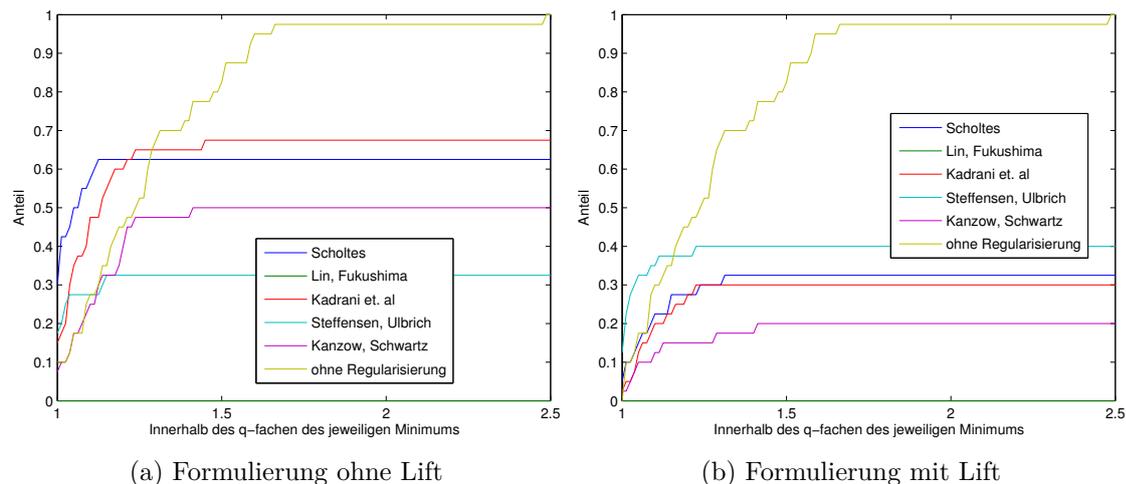


Abbildung E.1.: Vergleich der berechneten Zielfunktionswerte

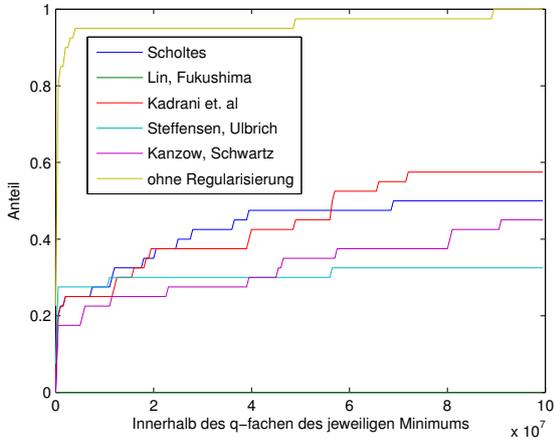
Vergleicht man diese Ergebnisse mit denen aus Kapitel 6, so stellt man, neben der leichten Verbesserung der Resultate des Regularisierungsverfahrens vor allem fest, dass das Verfahren von Steffensen und Ulbrich deutlich bessere Resultate liefert. Da diese Untersuchungen durchgeführt wurden, um den Ansatz von Steffensen und Ulbrich besser beurteilen zu können, sollen dessen Ergebnisse nun näher betrachtet werden.

Benutzt man die Problemformulierung ohne Lift so fällt auf, dass mit der Regularisierung von Steffensen und Ulbrich nun Probleme erfolgreich gelöst wurden. Gleichzeitig blieben sowohl der Anteil der gelösten Probleme als auch die berechnete Genauigkeit

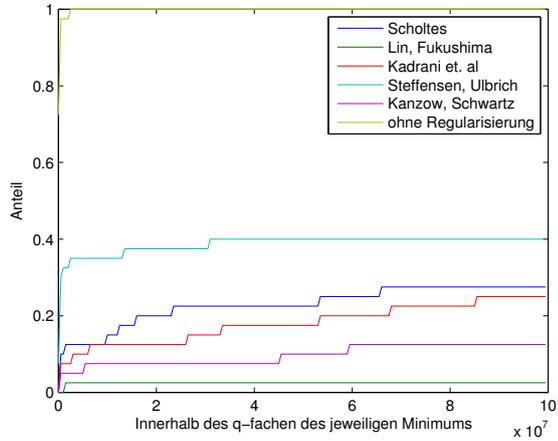
hinter denen der Regularisierungen von Scholtes, Kadrani et al. sowie von Kanzow und Schwartz zurück. Dies wird in den Abbildungen E.1a und E.2a deutlich. Darüber hinaus zeigt die Abbildung E.3a, dass die zur Berechnung benötigte Zeit vergleichbar mit den Zeiten ist, die die anderen Regularisierungen zur Berechnung der Ergebnisse benötigten. Bei der Problemformulierung mit Lift zeigte sich dagegen eine deutliche Verbesserung der numerischen Resultate gegenüber den Ergebnissen der ursprünglichen Probleme. So wird in Abbildung E.1b deutlich, dass mit der Regularisierung von Steffensen und Ulbrich der Anteil der gelösten Probleme höher als bei den anderen Regularisierungen war. Gleichzeitig tritt hier der Fall ein, dass die Regularisierung von Steffensen und Ulbrich bei der Problemformulierung mit Lift mehr Probleme gelöst werden konnten, als bei der Nutzung der Formulierung ohne Lift. Auch beim Vergleich der erreichten Genauigkeiten erzielte die Regularisierung von Steffensen und Ulbrich geringfügig bessere Resultate als die übrigen Regularisierungen (vgl. Abbildung E.2b). Die zur Berechnung benötigte Zeit war bei der Verwendung der Regularisierung nach Steffensen und Ulbrich am kürzesten, sodass auch nach diesem Kriterium das beste Ergebnis mit der Regularisierung nach Steffensen und Ulbrich erzielt wurde, was sich in Abbildung E.3b zeigt.

Insgesamt kann man somit sagen, dass durch die betrachtete Modifizierung der Probleme die Ergebnisse der Regularisierung nach Steffensen und Ulbrich deutlich bessere Resultate lieferte. Darüber hinaus erhält man hier, dass für diese Regularisierung die Ergebnisse bei der Verwendung der Formulierung mit bzw. ohne Lift vergleichbar waren. Während bei der Formulierung ohne Lift die Ergebnisse schneller berechnet wurden, war der Anteil der gelösten Problem bei der Formulierung mit Lift etwas höher.

Abschließend muss jedoch noch angemerkt werden, dass die hier gelösten Probleme letztlich bedeuten, dass auf jeder Kante eine Maut der Mindesthöhe 1 eingeführt wird. Auch wenn diese Fälle bei praktischen Anwendungen vorkommen können, so stellt dies doch eine erhebliche Einschränkung der Allgemeingültigkeit des eingeführten Modells dar.

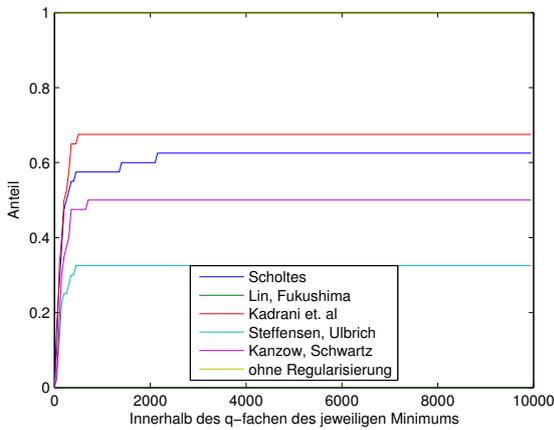


(a) Formulierung ohne Lift

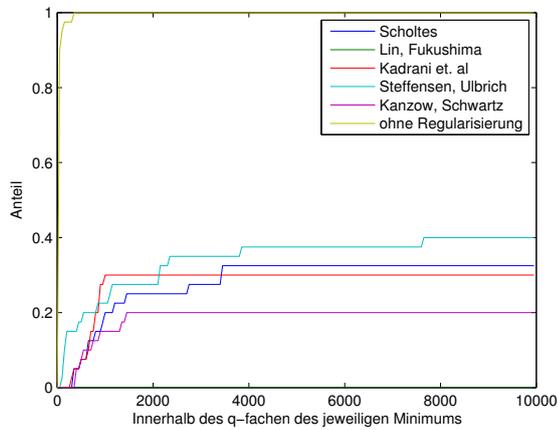


(b) Formulierung mit Lift

Abbildung E.2.: Vergleich der Verletzung der Nebenbedingungen



(a) Formulierung ohne Lift



(b) Formulierung mit Lift

Abbildung E.3.: Vergleich der benötigten Zeit